Computing Puiseux series: a fast divide and conquer algorithm

Adrien Poteaux, Martin Weimann

Computing Puiseux series: a fast divide and conquer algorithm

Adrien Poteaux,
CRIStAL, Université de Lille
UMR CNRS 9189, Bâtiment M3
59655 Villeneuve d'Ascq, France
`adrien.poteaux@univ-lille.fr`

Martin Weimann
GAATI*, Université de Polynésie Française
UMR CNRS 6139, BP 6570
98702 Faa'a, Polynésie Française
`martin.weimann@upf.pf`

December 3, 2018

Let $F \in \mathbb{K}[X,Y]$ be a polynomial of total degree D defined over a perfect field $\mathbb{K}$ of characteristic zero or greater than D. Assuming F separable with respect to Y, we provide an algorithm that computes all Puiseux series of F above $X = 0$ in less than $\tilde{\mathcal{O}}(D\,\delta)$ operations in $\mathbb{K}$, where δ is the valuation of the resultant of F and its partial derivative with respect to Y. To this aim, we use a divide and conquer strategy and replace univariate factorisation by dynamic evaluation. As a first main corollary, we compute the irreducible factors of F in $\mathbb{K}[[X]][Y]$ up to an arbitrary precision X^N with $\tilde{\mathcal{O}}(D(\delta+N))$ arithmetic operations. As a second main corollary, we compute the genus of the plane curve defined by F with $\tilde{\mathcal{O}}(D^3)$ arithmetic operations and, if $\mathbb{K} = \mathbb{Q}$, with $\tilde{\mathcal{O}}((h+1)D^3)$ bit operations using probabilistic algorithms, where h is the logarithmic height of F.

*Current delegation. Permanent position at LMNO, University of Caen-Normandie, BP 5186, 14032 Caen Cedex, France.

1 Introduction.

This paper provides complexity results for computing Puiseux series of a bivariate polynomial with coefficients over a perfect field of characteristic zero or big enough.

Context and main results. In this paper, $\mathbb{K}$ denotes a perfect field (e.g. $\mathbb{K}$ iss a finite or number field), p its characteristic, X and Y two indeterminates over $\mathbb{K}$ and $F \in \mathbb{K}[X,Y]$ a bivariate polynomial primitive and separable in Y. We denote D the total degree of F, $d_X = \deg_X(F)$ and $d_Y = \deg_Y(F)$; we always assume $p = 0$ or $p > d_Y$. Let $\overline{\mathbb{K}}$ be the algebraic closure of $\mathbb{K}$ and $\delta = \upsilon_X(R_F)$ the X-valuation of the resultant $R_F = \mathrm{Res}_Y(F, F_Y)$ of F and its Y-derivative F_Y. With our assumption on p, the Puiseux theorem states that for any $x_0 \in \overline{\mathbb{K}}$, the roots of F (viewed as a univariate polynomial in Y) may be expressed as fractional Laurent power series in $(X - x_0)$ with coefficients in $\overline{\mathbb{K}}$. These are the (classical) *Puiseux series*[1] of F above x_0, fundamental objects of the theory of algebraic curves [8, 39]. Many applications are given in [32, 33].

For the computation of *singular parts* of Puiseux series (that contain the relevant information about the singularities of the associated curve; remaining terms can be computed up to an arbitrary precision in quasi-linear time via Newton iterations), we get:

Theorem 1. *There exists an algorithm*[2] *that computes singular parts of Puiseux series of F above $x_0 = 0$ in an expected $\tilde{\mathcal{O}}(d_Y\,\delta)$ arithmetic operations over $\mathbb{K}$.*

Here we use the classical $\tilde{\mathcal{O}}$ notation that omits logarithmic factors (see Section 2.3). This improves the bound $\tilde{\mathcal{O}}({d_Y}^2\,\delta)$ of [33]. From that we deduce:

Theorem 2. *There exists an algorithm that computes the singular part of Puiseux series of F above* all *critical points in an expected $\tilde{\mathcal{O}}({d_Y}^2 d_X) \subset \tilde{\mathcal{O}}(D^3)$ arithmetic operations.*

This improves the bound $\tilde{\mathcal{O}}({d_Y}^2 {d_X}^3) \subset \tilde{\mathcal{O}}(D^5)$ of [30, 31]; note that [33, Proposition 12] suggests a bound $\tilde{\mathcal{O}}({d_Y}^3 d_X) \subset \tilde{\mathcal{O}}(D^4)$. Via the Riemann-Hurwitz formula, we get:

Corollary 1. *Assuming $p = 0$ or $p > D$, there exists an algorithm that computes the genus of a given geometrically irreducible algebraic plane curve over $\mathbb{K}$ of degree D in an expected $\tilde{\mathcal{O}}(D^3)$ arithmetic operations.*

Moreover, using the reduction criterion of [30, 32], we can bound the bit complexity of the genus computation (here $\mathrm{ht}(P)$ stands for the maximum between the logarithm of the denominator of P, and the logarithm of the infinite norm of its numerator):

Corollary 2. *Let $\mathbb{K} = \mathbb{Q}(\gamma)$ be a number field, $0 < \epsilon < 1$ a real number and $F \in \mathbb{K}[X,Y]$. Denote M_γ the minimal polynomial of γ and w its degree. Then there exists a Monte*

[1] terms written *in italics* in this introduction are defined in Section 2 or 5.1.

[2] our algorithms are Las Vegas, due to the computation of primitive elements; they should become deterministic via the preprint [38]. See Remark 3 and Sections 3.1 and 5.2

Carlo algorithm that computes the genus of the curve $F(X,Y) = 0$ with probability of error less than ϵ and an expected number of word operations in:

$$\tilde{\mathcal{O}}({d_Y}^2 d_X w^2 \log^2 \epsilon^{-1}[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F) + 1]).$$

With the same notations as in Corollary 2, we have:

Corollary 3. *Assuming that the degree of the square-free part of the resultant $Res_Y(F, F_Y)$ is known, there exists a Las Vegas algorithm that computes the genus of the curve $F(X,Y) = 0$ with an expected number of word operations in:*

$$\tilde{\mathcal{O}}({d_Y}^2 d_X w^2[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F) + 1]).$$

Finally, our algorithm induces a fast analytic factorisation of F:

Theorem 3. *There exists an algorithm that computes the irreducible analytic factors of F in $\mathbb{K}[[X]][Y]$ with precision $N \in \mathbb{N}$ in an expected $\tilde{\mathcal{O}}(d_Y(\delta+N))$ arithmetic operations in $\mathbb{K}$, plus the cost of one univariate factorisation of degree at most d_Y.*

This has a particular interest with regards to factorisation in $\mathbb{K}[X,Y]$ or $\overline{\mathbb{K}}[X,Y]$: when working along a critical fiber, one can take advantage of some combinatorial constraints imposed by ramification when recombining analytic factors into rational factors [40].

Main ideas and organisation of the paper. Classical definitions related to Puiseux series and description of the *rational Newton–Puiseux algorithm* of [14] are provided in Section 2. Then, the paper is organised accordingly to the following main ideas:

Idea 1. Concentrate on the monic case. The roots above $(0,\infty)$ require special care (see Section 4.5), This is why we use $\delta = v_X(\mathrm{Res}_Y(F, F_Y))$ and not $v_X(\mathrm{Disc}_Y(F)) \leq \delta$.

Idea 2. Use tight truncation bounds for the powers of X in the course of the algorithm. The bound $n = \delta$ can be reached for *some* Puiseux series, but we prove in Section 3 that we can compute at least half of them using a bound $n \in \mathcal{O}(\delta/d_Y)$.

Idea 3. A divide and conquer algorithm. From Idea 2, we prove that F is irreducible (and get its Puiseux series) or get a factorisation $F = G\,H \mod X^n$ where $n \in \mathcal{O}(\delta/d_Y)$, G corresponds to the computed Puiseux series, and H satisfies $\deg_Y(H) \leq d_Y/2$. The fiber $X = 0$ being critical, $G(0,Y)$ and $H(0,Y)$ are not coprime, and the classical Hensel lemma does not apply. But it can be adapted to our case to lift the factorisation $F = G\,H$ up to precision δ. This requires a Bézout relation $U\,G + V\,H = X^\kappa$ with $\kappa \in \mathcal{O}(\delta/d_Y)$, computed via [24]. Finally, we recursively compute the Puiseux series of H, defining a divide and conquer algorithm to compute an analytic factorisation of $F \mod X^{\delta+1}$, together with the singular parts of its Puiseux series above $x_0 = 0$. See Section 4.

Idea 4. We rely on dynamic evaluation. The next step is to get rid of univariate factorisations, which are too expansive for our purpose. In Section 5, we use dynamic evaluation [12, 13] to avoid this bottleneck, leading to work over product of fields: we have to pay attention to zero divisors and perform suitable splittings when required.

These ideas allow us to compute the desingularisation of the curve above all its critical points in Section 6. We get a complexity bound, as good as, up to logarithmic factors, the best known algorithm to compute bivariate resultants. This is Theorem 2.

Finally, we develop a fast factorisation algorithm and prove Theorem 3 in Section 7.

To conclude, we add further remarks in Section 8, showing in particular that any Newton–Puiseux like algorithm would not lead to a better worst case complexity.

A brief state of the art. In [14], D. Duval defines the rational Newton–Puiseux algorithm over a field $\mathbb{K}$ with characteristic 0. From the complexity analysis therein, it takes less than $\mathcal{O}({d_Y}^6\,{d_X}^2)$ operations in $\mathbb{K}$ when F is monic (no fast algorithm is used). This algorithm uses the D5-principle, and can trivially be generalised when $p > d_Y$.

In [30, 31], an algorithm with complexity $\tilde{\mathcal{O}}(d_Y\,\delta^2 + d_Y\,\delta\,\log(p^c))$ is provided over $\mathbb{K} = \mathbb{F}_{p^c}$, with $p > d_Y$. From this bound is deduced an algorithm that computes the singular parts of Puiseux series of F above *all* critical points in $\tilde{\mathcal{O}}({d_Y}^3\,{d_X}^2\,\log(p^c))$. In [33], still considering $\mathbb{K} = \mathbb{F}_{p^c}$, an algorithm is given to compute the singular part of Puiseux series over $x_0 = 0$ in an expected $\tilde{\mathcal{O}}(\rho\,d_Y\,\delta + \rho\,d_Y\,\log(p^c))$ arithmetic operations, where ρ is the number of rational Puiseux expansions above $x_0 = 0$ (bounded by d_Y). These two algorithms use univariate factorisation over finite fields, thus cannot be directly extended to the 0 characteristic case. This also explains why the second result does not provide an improved bound for the computation of Puiseux series above *all* critical points.

There are other methods to compute Puiseux series or analytic factorisation, as generalised Hensel constructions [3, 20], or the Montes algorithm [4, 28] (which works over general local fields). Several of these methods and a few others have been commented in previous papers by the first author [32, 33]. Also, there exist algorithms for the genus based on linear differential operators and avoiding the computation of Puiseux series [6, 26]. To our knowledge, none of these methods have been proved to provide a complexity which fits in the bounds obtained in this paper.

Acknowledgment. This paper is dedicated to Marc Rybowicz, who passed away in November 2016 [15]. The first ideas of this paper actually came from a collaboration between Marc and the first author in the beginning of 2012, that led to [33] as a first step towards the divide and conquer algorithm presented here. We also thank François Lemaire for many useful discussions on dynamic evaluation.

2 Main definitions and classical algorithms.

2.1 Puiseux series.

We keep notations of Section 1. Up to a change of variable $X \leftarrow X + x_0$, it is sufficient to give definitions and properties for the case $x_0 = 0$. Under the assumption that $p = 0$

or $p > d_Y$, the well known Puiseux theorem asserts that the d_Y roots of F (viewed as a univariate polynomial in Y) lie in the field of Puiseux series $\cup_{e\in\mathbb{N}}\overline{\mathbb{K}}((X^{1/e}))$. See [8, 16, 39] or most textbooks about algebraic functions for the 0 characteristic case. When $p > d_Y$, see [11, Chap. IV, Sec. 6]. It happens that these Puiseux series can be grouped according to the field extension they define. Following Duval [14, Theorem 2], we consider decompositions into irreducible elements:

$$\begin{aligned} F &= \prod_{i=1}^{\rho} F_i \text{ with } F_i \text{ irreducible in } \mathbb{K}[[X]][Y] \\ F_i &= \prod_{j=1}^{f_i} F_{ij} \text{ with } F_{ij} \text{ irreducible in } \overline{\mathbb{K}}[[X]][Y] \\ F_{ij} &= \prod_{k=0}^{e_i-1} \left(Y - S_{ij}(X^{1/e_i}\zeta_{e_i}^k)\right) \text{ with } S_{ij} \in \overline{\mathbb{K}}((X)) \end{aligned}$$

with $\zeta_{e_i} \in \overline{\mathbb{K}}$ is a primitive e_i-th root of unity. Primitive roots are chosen so that $\zeta_{ab}^b = \zeta_a$.

Definition 1. The d_Y fractional Laurent series $S_{ijk}(X) = S_{ij}(X^{1/e_i}\zeta_{e_i}^k) \in \overline{\mathbb{K}}((X^{1/e_i}))$ are called the *classical Puiseux series* of F above 0. The integer $e_i \in \mathbb{N}$ is the *ramification index* of S_{ij}. If $S_{ij} \in \overline{\mathbb{K}}[[X^{1/e_i}]]$, we say that S_{ij} *is defined at* $x_0 = 0$.

Proposition 1. *The* $\{F_{ij}\}_{1\leq j\leq f_i}$ *have coefficients in a degree* f_i *extension* $\mathbb{K}_i$ *of* $\mathbb{K}$. *They are conjugated by the action of the Galois group of* $\mathbb{K}_i/\mathbb{K}$. *We call* $\mathbb{K}_i$ *the* residue field *of any Puiseux series of* F_i *and* f_i *its* residual degree. *We have the relation* $\sum_{i=1}^{\rho} e_i f_i = d_Y$.

Proof. First claim is [14, Section 1]. Second one is e.g. [11, Chapter 4, Section 1]. □

This leads to the definition of rational Puiseux expansions (classical Puiseux series can be constructed from a system of rational Puiseux expansions - see e.g. [33, Section 2]):

Definition 2. A system of *rational Puiseux expansions* over $\mathbb{K}$ ($\mathbb{K}$-RPE) of F above 0 is a set $\{R_i\}_{1\leq i\leq \rho}$ such that:

- $R_i(T) \in \mathbb{K}_i((T))^2$;
- $R_i(T) = (X_i(T), Y_i(T)) = \left(\gamma_i T^{e_i}, \sum_{l=n_i}^{\infty} \beta_{il} T^l\right)$, with $n_i \in \mathbb{Z}$, $\gamma_i \neq 0$ and $\beta_{i,n_i} \neq 0$;
- R_i is a parametrisation of F_i, i.e. $F_i(X_i(T), Y_i(T)) = 0$;
- the parametrisation is irreducible, i.e. e_i is minimal.

We call $(X_i(0), Y_i(0))$ the *center* of R_i. We have $Y_i(0) = \infty$ if $n_i < 0$, which happens only for non monic polynomials.

Throughout this paper, we will truncate the powers of X of polynomials or series. To that purpose, we introduce the following notation: given $\tau \in \mathbb{Q}$ and a Puiseux series $S = \sum_{\alpha\in\mathbb{Q}} c_\alpha X^\alpha$, we denote $\lceil S\rceil^\tau = \sum_{\alpha\leq\tau} \alpha X^\alpha$ (this sum having thus a finite number of terms). We generalize this notation to polynomials with coefficients in the field of Puiseux series by applying it coefficient-wise. In particular, if $H \in \mathbb{K}[[X]][Y]$ is defined as $H = \sum_i (\sum_{k \geq 0} \alpha_{ik} X^k) Y^i$, then $\lceil H \rceil^\tau = \sum_i (\sum_{k=0}^{\lfloor \tau \rfloor} \alpha_{ik} X^k) Y^i$.

Definition 3. The *regularity index* r of a Puiseux series S of F with ramification index e is the least integer $N \geq \min(0, e\, \upsilon_X(S))$ such that, if $\lceil S \rceil^{\frac{N}{e}} = \lceil S' \rceil^{\frac{N}{e}}$ for some Puiseux series S' of F, then $S = S'$. We call $\lceil S \rceil^{\frac{r}{e}}$ the *singular part* of S in F.

Roughly speaking, the regularity index is the number of terms necessary to "separate" a Puiseux series from all the others (with a special care when $\upsilon_X(S) < 0$).

Example 1. Consider $F_1 \in \mathbb{F}_{29}[X, Y]$ defined as $F_1 = \prod_{i=1}^{3}(Y - S_i(X)) + X^{19} Y$ with $S_i = X + X^2 + X^3 + 17\, X^4 + X^5 + X^6 + X^7 + (-1)^i X^{15/2}$, $1 \leq i \leq 2$ and $S_3 = X + X^2 + X^3 + X^4$. The singular parts of the Puiseux series of F_1 are precisely the S_i, with regularity indices respectively $r_1 = r_2 = 15$ and $r_3 = 4$.

Since regularity indices of all Puiseux series corresponding to the same rational Puiseux expansion are equal, we define:

Definition 4. The *singular part* of a rational Puiseux expansion R_i of F is the pair

$$\left(\gamma_i T^{e_i}, \Gamma(T) = \sum_{k=n_i}^{r_i} \beta_{ik} T^k\right),$$

where r_i is the regularity index of R_i, i.e. the one of any Puiseux series associated to R_i.

Once such a singular part has been computed, the implicit function theorem ensures us that one can compute the series up to an arbitrary precision. This can be done in quasi linear time by using a Newton operator [22, Corollaries 5.1 and 5.2, page 251].

Notations. In the remaining of the paper, we will denote $(R_i)_{1 \leq i \leq \rho}$ the rational Puiseux expansions of F. To any R_i, we will always associate the following notations:

- e_i, f_i and r_i will respectively be the ramification index, the residual degree and the regularity index of R_i,
- we define $v_i \in \mathbb{Q}$ as $\upsilon_X(F_Y(S))$ for any Puiseux series S associated to R_i.

Same notations will be used if S_i (or S_{ijk}) denotes a Puiseux series. If we omit any index i, we will use the notations e, f and r for the three first integers.

2.2 The rational Newton–Puiseux algorithm.

Our algorithm in Section 3 is a variant of the well known Newton–Puiseux algorithm [8, 39]. We now explain (roughly speaking) the idea of this algorithm via an example, and then describe the variant of D. Duval [14, section 4] (we use its improvements).

Tools and idea of the algorithm. Let $F_0(X,Y) = Y^6 + Y^5X + 5\,Y^4X^3 - 2\,Y^4X + 4\,Y^2X^2 + X^5 - 3\,X^4$ and consider its Puiseux series computation. From the Puiseux theorem, the first term of any such series $S(X)$ is $\alpha\, X^{\frac{m}{q}}$ for some $\alpha \in \overline{\mathbb{K}}$ and $(m,q) \in \mathbb{N}^2$. We have $F_0(X, \alpha\, X^{\frac{m}{q}} + \cdots) = \alpha^6\, X^{\frac{6\,m}{q}} + \alpha^5\, X^{\frac{5\,m}{q}+1} + 5\,\alpha^4\, X^{\frac{4\,m}{q}+3} - 2\,\alpha^4\, X^{\frac{4\,m}{q}+1} + 4\,\alpha^2\, X^{\frac{2\,m}{q}+2} + X^5 - 3\,X^4 + \cdots$. To get $F_0(X, S(X)) = 0$, at least two terms of the previous sum must cancel one another, i.e. (m,q) must be chosen so that two or more of the exponents coincide. To that purpose, we use the following definition:

Definition 5. The *support* of $F = \sum_{i,j} \alpha_{ij} X^j\, Y^i$ is the set $\{(i,j) \in \mathbb{N}^2 \mid \alpha_{ij} \neq 0\}$.

Note that the powers of Y are given by the horizontal axis. The condition on (m,q) can be translated as: two points of the support of F_0 belong to the same line $m\,a + q\,b = l$. To increase the X-order of the evaluation, no point must be under this line. Here we have two such lines, $a + 2\,b = 6$ and $a + b = 4$, that define the Newton polygon of F_0:

Definition 6. The *Newton polygon* $\mathcal{N}(F)$ of F is the lower part of the convex hull of its support.

We are now considering the choice of α corresponding to $a + 2\,b = 6$. We have $F_2(T^2, \alpha\, T) = (\alpha^6 - 2\alpha^4 + 4\alpha^2)\, T^6 - 3\,T^8 + \alpha^5\, T^7 + (5\alpha^4 + 1)T^{10} + \ldots$, meaning that α must be a non zero root of $P(Z) = Z^6 - 2\,Z^4 + 4Z^2$. Then, to get more terms, we recursively apply this strategy to the polynomial $F_2(X^2, X\,(Y+\alpha))$. Actually, it is more interesting to consider a root $\xi = \alpha^2$ of the polynomial $\phi(Z) = Z^2 - 2\,Z + 4$ (we have $P(Z) = Z^2\, \phi(Z^2)$ and we are obviously not interested in the root $\alpha = 0$), which is the characteristic polynomial [14]:

Definition 7. If $F = \sum \alpha_{ij} X^j Y^i$, then the *characteristic polynomial* ϕ_Δ of $\Delta \in \mathcal{N}(F)$ is $\phi_\Delta(T) = \sum_{(a,b)\in\Delta} \alpha_{ab} T^{\frac{a-a_0}{q}}$ where a_0 is the smallest value such that (a_0, b_0) belongs to Δ for some b_0.

Description of the algorithm. We now provide a formal definition of the `RNPuiseux` algorithm for monic polynomials (see Section 4.5 for the non monic case); it uses two sub algorithms, for each we only provide specifications, and an additional definition, the *modified* Newton polygon [33, Definition 6]. The latter enables `RNPuiseux` to output *precisely* the singular part. We will not use it in our strategy, except for the proof of Lemma 5 (see Remark 6). For the sake of completness, we recall it below.

- If $F = \sum_{i=0}^{d_Y} \alpha_i(X)\, Y^i$, the *modified Newton polygon* $\mathcal{N}^\star(H)$ is constructed as follow: if $\alpha_0 = 0$ (resp. $\alpha_0 \neq 0$ and the first edge, starting from the left, ends at $(1, v_X(\alpha_1))$), add to $\mathcal{N}(F)$ (resp. replace the first edge by) a fictitious edge joining the vertical axis to $(1, v_X(\alpha_1))$ such that its slope is the largest (negative or null) integer less than or equal to the slope of the next edge (see Figure 1a).
- `Bézout`, given $(q,m) \in \mathbb{Z}^2$ with $q > 0$, computes $(u,v) \in \mathbb{Z}^2$ s.t. $u\,q - m\,v = 1$ and $0 \leq v < q$.

- `Factor`, given $\mathbb{K}$ a field and ϕ a univariate polynomial over $\mathbb{K}$, computes the factorisation of ϕ over $\mathbb{K}$, given as a list of factors and multiplicities.

Algorithm: `RNPuiseux`$(F, \mathbb{K}, \pi)$

In: $F \in \mathbb{K}[X,Y]$ monic, $\mathbb{K}$ a field and π the result of previous computations ($\pi = (X,Y)$ for the initial call)

Out: A set of singular parts of rational Puiseux expansions above $(0,0)$ of F with their base field.

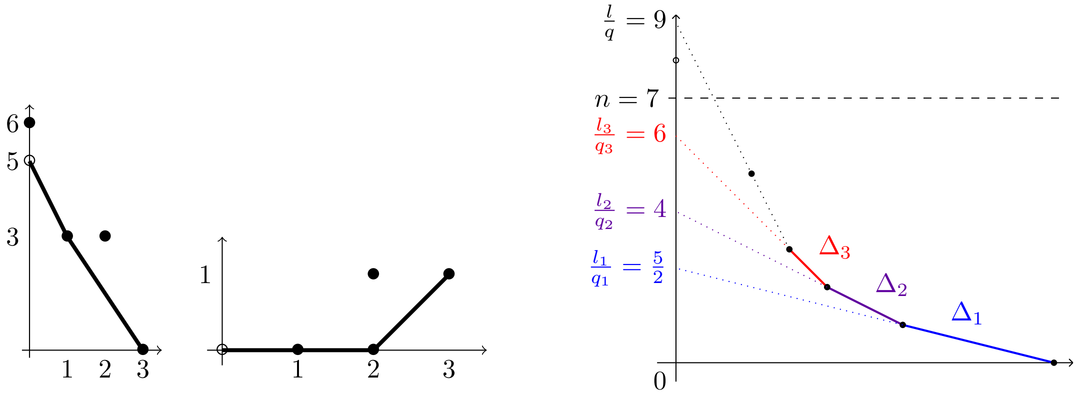
1 $\mathcal{R} \leftarrow \{\}$; `// results of the algorithm will be grouped in` $\mathcal{R}$
2 **foreach** $\Delta \in \mathcal{N}^{\star}(F)$ **do** `// we consider only negative slopes`
3   Compute m, q, l, ϕ_Δ associated to Δ;
4     $(u,v) \leftarrow$ `Bézout`(m,q);
5   **foreach** (ϕ, M) **in** `Factor`(ϕ_Δ) **do**
6   Take ξ a new symbol satisfying $\phi(\xi) = 0$;
7         $\pi_1 = \pi(\xi^v X^q, X^m (Y + \xi^u))$;
8   **if** $M = 1$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\pi_1(T,0), \mathbb{K}(\xi))\}$;
9   **else**
10             $H(X,Y) \leftarrow F(\xi^v X^q, X^m (Y + \xi^u))/X^l$; `// Puiseux transform`
11             $\mathcal{R} \leftarrow \mathcal{R} \cup$ `RNPuiseux`$(H, \mathbb{K}(\xi), \pi_1)$;
12 **return** $\mathcal{R}$;

The key improvement of this rational version is the distribution of ξ to both X and Y variables (line 10). This avoids to work with $\alpha = \xi^{1/q}$ and to introduce any useless field extension due to ramification (see [14, Section 4]).

Truncated Newton polygon. In this paper, we will use low truncation bounds; in particular, we may truncate some points of the Newton polygon. In order to certify the correctness of the computed slopes, we will use the following definition:

Definition 8. Given $F \in \mathbb{K}[X,Y]$ and $n \in \mathbb{N}$, the *n-truncated Newton polygon* of F is the set $\mathcal{N}_n(F)$ composed of edges Δ of $\mathcal{N}(\lceil F \rceil^n)$ that satisfy $\frac{l}{q} \leq n$ if Δ belongs to the line $m\,a + q\,b = l$. In particular, any edge of $\mathcal{N}_n(F)$ is an edge of $\mathcal{N}(F)$.

Example 2. Let us consider $F_2 = Y^{10} + X\,Y^6 + X^2\,Y^4 + X^3\,Y^3 + X^5\,Y^2 + X^8$ and $n = 7$. Figure 1b provides the truncated Newton polygon of F_2 with precision 7. Here we have $\lceil F_2 \rceil^7 = F_2 - X^8$ and $\mathcal{N}(\lceil F_2 \rceil^7) = [(10,0),(6,1),(4,2),(3,3),(2,5)]$. But the edge $[(3,3),(2,5)]$ is not part of $\mathcal{N}_7(F_2)$, as it belongs to $2\,a + b = 9$, and that there are points (i,j) so that $2\,i + j \leq 9$ and $j > 7$: from the knowledge of $\lceil F_2 \rceil^7$, we cannot guarantee that $\mathcal{N}(F_2)$ contains an edge belonging to $2\,a + b = 9$. This is indeed wrong here, since $\mathcal{N}(F_2) = [(10,0),(6,1),(4,2),(3,3),(0,8)]$.



(a) $\mathcal{N}^\star(F)$ for resp. $F = (Y - X^3)\,(Y^2 - X^3)$ and $F = Y\,(Y-1)\,(XY-1)$

(b) $\mathcal{N}_7(F_2)$ for $F_2 = Y^{10} + X\,Y^6 + X^2\,Y^4 + X^3\,Y^3 + X^5\,Y^2 + X^8$

Figure 1: The modified and truncated Newton polygons

2.3 Complexity model.

In this paper, we use two model of computations ; both are RAM models: the algebraic RAM of Kaltofen [21, Section 2] and the boolean one. The latter is considered only for Corollaries 2 and 3, where we just estimate word operations generated by arithmetic operations in various coefficient fields (assuming for instance a constant time access to coefficients of polynomials). For the arithmetic model, we only count the number of arithmetic operations (addition, multiplication, division) in our base field $\mathbb{K}$. Most subalgorithms are deterministic; for them, we consider the worst case. However, computation of primitive elements uses a probabilistic of Las Vegas type algorithm. Their running times depend on random choices of element in $\mathbb{K}$; hence, we use average running times, that propagate to our main results.

Our complexity results use the classical notations $\mathcal{O}()$ and $\tilde{\mathcal{O}}()$ that respectively hide constant and logarithmic factors. See for instance [18, Chapter 25, Section 7].

Polynomial multiplication. We finally recall some classical complexity results, starting with the multiplication of univariate polynomials:

Definition 9. A (univariate) *multiplication time* is a map $\mathsf{M} : \mathbb{N} \to \mathbb{R}$ such that:

- for any ring $\mathbb{A}$, polynomials of degree less than d in $\mathbb{A}[X]$ can be multiplied in at most $\mathsf{M}(d)$ operations (multiplication or addition) in $\mathbb{A}$;
- for any $0 < d \le d'$, the inequality $\mathsf{M}(d)\,d' \le \mathsf{M}(d')\,d$ holds.

Lemma 1. *Let* M *be a multiplication time. Then we have:*

1. $\mathsf{M}(d + d') \ge \mathsf{M}(d) + \mathsf{M}(d')$ *for any* $d, d' \in \mathbb{N}$,
2. $\mathsf{M}(1) + \mathsf{M}(2) + \cdots + \mathsf{M}(2^{k-1}) + \mathsf{M}(2^k) \le \mathsf{M}(2^{k+1})$ *for any* $k \in \mathbb{N}$.

Proof. The first point is [18, Exercise 8.33]. The second one is a direct consequence. $\square$

The best known multiplication time gives $\mathsf{M}(d) \in \mathcal{O}(d \log(d) \log(\log(d))) \subset \mathcal{O}\tilde{}(d)$ [9, 35]. Note in particular that for this value of $\mathsf{M}()$, we do not have $\mathsf{M}(d)\,\mathsf{M}(d') \leq \mathsf{M}(d\,d')$ but only $\mathsf{M}(d)\,\mathsf{M}(d') \leq \mathsf{M}(d\,d') \log(d\,d')$. This is why we use Kronecker substitution.

Multiplication of multivariate polynomials. Consider two polynomials belonging to $\mathbb{A}[Z_1, \cdots, Z_s]$. Denote d_i a bound for their degrees in Z_i. Then, by Kronecker substitution, they can be multiplied in less than $\mathcal{O}(\mathsf{M}(2^{s-1}\, d_1 \cdots d_s))$ operations in $\mathbb{A}$ (it is straightforward to adapt [18, Corollary 8.28, page 247] to any number of variables). In particular, if s is constant, the complexity bound is $\mathcal{O}(\mathsf{M}(d_1 \cdots d_s))$.

Bivariate polynomials defined over an extension of $\mathbb{K}$. Given an irreducible polynomial $P \in \mathbb{K}[Z]$, we denote $\mathbb{K}_P := \mathbb{K}[Z]/(P(Z))$ and $d_P := \deg_Z(P)$. In Sections 3 and 4, we multiply two polynomials in $\mathbb{K}_P[X, Y]$ as follows: first perform the polynomial multiplication over $\mathbb{K}[X, Y, Z]$ as stated in the previous paragraph; then apply the reduction modulo P on each coefficient. Denoting d_X (resp. d_Y) a bound for the degree in X (resp. Y) of the considered polynomials, the total cost is $\mathcal{O}(\mathsf{M}(d_X\, d_Y\, d_P))$ (see [18, Theorem 9.6, page 261] for the second point).

Matrix multiplication. Primitive elements computation are expressed via the well known $2 \leq \omega \leq 3$ exponent (so that one can multiply two square matrices of size d in less than $\mathcal{O}(d^\omega)$ operations over the base ring). We have $\omega < 2.373$ [23]. Note however that our results do not require fast matrix multiplication: they stand if we take $\omega = 3$.

Finally, note that we postpone the discussion concerning the complexity of operations modulo triangular sets (needed for dynamic evaluation) in Section 5.2.

3 Refined truncation bounds.

We keep notations of Sections 1, 2.1 and 2.3 ($\mathbb{K}_P$ and d_P). Additionally, we assume F to be monic. The aim of this section is to prove that we can compute at least half of the Puiseux series of F in less than $\mathcal{O}\tilde{}(d_Y\, \delta)$ arithmetic operations, not counting the factorisation of univariate polynomials. Our algorithms and intermediate results will use the following notion:

Definition 10. We say that $S_0 \in \overline{\mathbb{K}}((X^{1/e_0}))$ is a Puiseux series of F known with precision n if there exists a Puiseux series S of F s.t. $\lceil S_0 \rceil^n = \lceil S \rceil^n$. We say that $R_0 = (\gamma_0\, T^{e_0}, \Gamma_0(T))$ is a RPE of F known with precision n if $\lceil \Gamma_0((X/\gamma_0)^{1/e_0}) \rceil^n$ is a Puiseux series of F known with precision n.

Theorem 4. *There exists an algorithm that computes some RPEs $R_1, \cdots, R_\lambda$ of F known with precision at least $4\,\delta/d_Y$, containing their singular parts, and such that*

$\sum_{i=1}^{\lambda} e_i\, f_i \geq \frac{d_Y}{2}$. *Not taking into account univariate factorisations, this can be done in an expected* $\mathcal{O}(\mathsf{M}(d_Y\,\delta)\,\log(d_Y)) \subset \mathcal{O}\tilde{~}(d_Y\,\delta)$ *arithmetic operations over* $\mathbb{K}$.

Algorithm `Half-RNP` in Section 3.2 will be such an algorithm. It uses previous improvements by the first author and M. Rybowicz [30, 31, 33], and one additional idea, namely Idea 2 of Section 1.

3.1 Previous complexity improvements and Idea 2.

Lemma 2. *Let* $n \in \mathbb{N}$, $F \in \mathbb{K}_P[X,Y]$ *and* $\xi \in \mathbb{K}_P$ *for some irreducible* $P \in \mathbb{K}[Z]$. *Denote* Δ *an edge of* $\mathcal{N}(F)$ *belonging to* $m\,a{+}q\,b{=}l$, *and* $(u,v){=}\mathtt{Bézout}(m,q)$. *The Puiseux transform* $F(\xi^v X^q, X^m(\xi^u{+}Y))/X^l$ *modulo* X^n *can be computed as* n *univariate polynomial shifts over* $\mathbb{K}_P$. *It takes less than* $\mathcal{O}(n\,\mathsf{M}(d_Y\,d_P))$ *operations over* $\mathbb{K}$.

Proof. This is [31, Lemma 2, page 210]; Figure 2 illustrates the idea. Complexity also uses Kronecker substitution. □

Using the *Abhyankar's trick* [1, Chapter 12], we reduce the number of recursive calls of the rational Newton–Puiseux algorithm from δ to $\mathcal{O}(\rho\,\log(d_Y))$.

Lemma 3. *Let* $F = Y^{d_Y} + \sum_{i=0}^{d_Y-1} A_i(X)\,Y^i \in \mathbb{K}[X,Y]$ *with* $d_Y > 1$. *If the Newton polygon of* $F(X, Y - A_{d_Y-1}/d_Y)$ *has a unique edge* $(\Delta)\, m\,a + q\,b = l$ *with* $q = 1$, *then* ϕ_Δ *has several roots in* $\overline{\mathbb{K}}$.

In other words, after performing the Tschirnausen transform $Y \leftarrow Y - A_{d_Y-1}/d_Y$, we are sure to get at least either a branch separation, a non integer slope, or a non trivial factor of the characteristic polynomial. This happens at most $\mathcal{O}(\rho\,\log(d_Y))$ times.

Example 3. Let's consider once again the polynomial F_1 of Example 1. Its Newton polygon has a unique edge with integer slope, and the associated characteristic polynomial has a unique root. The Abhyankar's trick is applied with $\frac{1}{3}\,A_2 = X + X^2 + X^3 + 2\,X^4 + 20\,X^5 + 20\,X^6 + 20\,X^7$. Then, the shifted polynomial has still a unique edge, but its characteristic polynomial has two different roots: it separates S_3 from the two other Puiseux series.

Lemma 4. *Let* $F = Y^{d_Y} + \sum_{i=0}^{d_Y-1} A_i(X)\,Y^i \in \mathbb{K}_P[X,Y]$. *One can compute the truncated shift* $\lceil F(X, Y - A_{d_Y-1}/d_Y)\rceil^n$ *in less than* $\mathcal{O}(\mathsf{M}(n\,d_Y\,d_P))$ *operations over* $\mathbb{K}$.

Proof. From our assumption on the characteristic of $\mathbb{K}$, this computation can be reduced to bivariate polynomial multiplication via [5, Problem 2.6, page 15]. The result follows (see Section 2.3). □

In order to provide the monicity assumption of Lemma 3, the well-known Weierstrass preparation theorem [1, Chapter 16] is used.

Proposition 2. *Let $G \in \mathbb{K}_P[X,Y]$ not divisible by X. There exists unique $\widehat{G}$ and U in $\mathbb{K}_P[[X]][Y]$ s.t. $G = \widehat{G}\,U$, with $U(0,0) \neq 0$ and $\widehat{G}$ a Weierstrass polynomial of degree $\deg_Y(\widehat{G}) = \upsilon_Y(G(0,Y))$. Moreover, RPEs of G and $\widehat{G}$ centered at $(0,0)$ are the same.*

The following result provides a complexity bound.

Proposition 3. *Let $G \in \mathbb{K}_P[X,Y]$ as in Proposition 2 and $n \in \mathbb{N}$. Denote $\widehat{G}$ the Weierstrass polynomial of G. There exists an algorithm* `WPT` *that computes $\lceil \widehat{G} \rceil^n$ in less than $\mathcal{O}(\mathsf{M}(n \deg_Y(G)\, d_P))$ operations in $\mathbb{K}$.*

Proof. This is [18, Theorem 15.18, page 451], using Kronecker substitution for multivariate polynomial multiplication. This theorem assumes that $\mathrm{lc}_Y(G)$ is a unit, which is not necessarily the case here. However, formulæ in [18, Algorithm 15.10, pages 445 and 446] can still be applied in our context: this is exactly [25, Algorithm Q, page 33]. □

Representation of residue fields. As explained in [31, Section 5.1], representing residue fields as multiple extensions can be costly. Therefore, we need to compute primitive representations each time we get a characteristic polynomial ϕ with degree 2 or more. Note that algorithms we use here are Las-Vegas (this is the only probabilistic part concerning our results on Puiseux series computation).

Proposition 4. *Let $P \in \mathbb{K}[Z]$ and $\phi \in \mathbb{K}_P[W]$ be two irreducible polynomials of respective degrees $d_P = \deg_Z(P)$ and $d_\phi = \deg_W(\phi)$. Denote $d = d_P\, d_\phi$, and assume that there are at least d^2 elements in $\mathbb{K}$. There exists a Las-Vegas algorithm* `Primitive` *that computes an irreducible polynomial $P_1 \in \mathbb{K}[Z]$ with degree d together with an isomorphism $\Psi : \mathbb{K}_{P,\phi} \simeq \mathbb{K}_{P_1}$. It takes an expected $\mathcal{O}(d^{\frac{\omega+1}{2}})$ arithmetic operations plus a constant number of irreducibility tests in $\mathbb{K}[Z]$ of degree at most d. Moreover, given $\alpha \in \mathbb{K}_{P,\phi}$, one can compute $\Psi(\alpha)$ with $\mathcal{O}(d_P\, \mathsf{M}(d))$ operations over $\mathbb{K}$.*

Proof. See e.g. [34, Section 2.2]; some details are in the proof of Proposition 15. □

Remark 1. We do not precisely pay attention to the assumption about the number of elements in $\mathbb{K}$ in this paper. Note that we will always have $d \leq d_Y$ in our context. Therefore, if $\mathbb{K}$ is a finite field without enough elements, it is sufficient to build a degree 2 field extension since $p > d_Y$.

Remark 2. The above complexity result can actually be expressed as $\mathcal{O}(d^{\omega_0})$ where $\frac{3}{2} \leq \omega_0 \leq 2$ denotes an exponent so that one can multiply a $d \times \sqrt{d}$ matrix and a square $\sqrt{d} \times \sqrt{d}$ one with $\mathcal{O}(d^{\omega_0})$ operations in $\mathbb{K}$. One has $\omega_0 < 1.667$ from [19], which is better than the best known bound $\frac{\omega+1}{2} < 1.687$ [23]. This however does not improve our main results, since we could take $\omega = 3$ for our results to stand.

Remark 3. [38, Section 4] provides an almost linear deterministic algorithm to compute modulo tower of fields by computing "accelerated towers" instead of primitive elements. Such a strategy would lead to a version of Theorem 4 with a deterministic algorithm and a complexity bound $\mathcal{O}({d_Y}^{1+o(1)}\, \delta)$. Their preprint does not however deal with dynamic evaluation, so this can not be directly be used in Section 5, thus in our main results.

3.2 The `Half-RNP` algorithm.

We detail the algorithm mentionned in Theorem 4. It computes *truncated parametrisations* of F, i.e. maps $\pi = (\gamma\, X^e, \Gamma(X) + \alpha\, X^\tau\, Y)$ s.t. $\pi(T, 0)$ is a RPE of F known with precision $\frac{\tau}{e}$ (see Definition 10). Except possibly at the first call, H therein is Weierstrass.

Algorithm: `Half-RNP`(H, P, n, π)

In: $P \in \mathbb{K}[Z]$ irreducible, $H \in \mathbb{K}_P[X, Y]$ separable and monic in Y with $d := \deg_Y(H) > 0$, $n \in \mathbb{N}$ (truncation order) and π the current truncated-parametrisation ($P = Z$ and $\pi = (X, Y)$ for the initial call).

Out: all RPEs R_i of H s.t. $n - v_i \geq r_i$, with precision $(n - v_i)/e_i \geq r_i/e_i$.

1 $\mathcal{R} \leftarrow \{\}$; $B \leftarrow A_{d-1}/d$; $\pi_1 \leftarrow \lceil \pi(X, Y - B) \rceil^n$; // $H = \sum_{i=0}^{d} A_i Y^i$

2 **if** $d = 1$ **then return** $\pi_1(T, 0)$ **else** $H_1 \leftarrow \lceil H(X, Y - B) \rceil^n$;

3 **foreach** Δ **in** $\mathcal{N}_n(H_1)$ **do** // Δ belongs to $m\,a + q\,b = l$

4   **foreach** (ϕ, M) **in** `Factor`$(\mathbb{K}_P, \phi_\Delta)$ **do**

5   **if** $\deg_W(\phi) = 1$ **then** $\xi, P_1, H_2, \pi_2 = -\phi(Z, 0), P, H_1, \pi_1$;

6   **else**

7             $(P_1, \Psi) \leftarrow$ `Primitive`(P, ϕ);

8             $\xi, H_2, \pi_2 \leftarrow \Psi(W), \Psi(H_1), \Psi(\pi_1)$; // $\Psi : \mathbb{K}_{P,\phi} \to \mathbb{K}_{P_1}$ isomorphism

9         $\pi_3 \leftarrow \pi_2(\xi^v\, X^q, X^m\, (Y + \xi^u)) \mod P_1$; // $u, v =$ `Bézout`(m, q)

10         $H_3 \leftarrow \lceil H_2(\xi^v\, X^q, X^m\, (Y + \xi^u)) \rceil^{n_1} \mod P_1$; // $n_1 = q\, n - l$

11         $H_4 \leftarrow$ `WPT`(H_3, n_1);

12         $\mathcal{R} \leftarrow \mathcal{R} \cup$ `Half-RNP`(H_4, P_1, n_1, π_3)

13 **return** $\mathcal{R}$;

Remark 4. We have $\deg_X(\pi) \leq n\, e_i$ for any RPE deduced from π. This is obvious when π is defined from Line 1; changing X by X^q on Line 9 is also straightforward. Also, we have $m \leq n\, e_i$, since $\frac{m}{q} \leq \frac{l}{q} \leq n$ from Definition 8.

Theorem 4 is an immediate consequence of the following result, proved in Section 3.4.

Proposition 5. `Half-RNP`$(F, Z, 6\,\delta/d_Y, (X, Y))$ *outputs a set of RPEs. Among them is a set* $R_1, \cdots, R_\lambda$ *known with precision at least* $4\,\delta/d_Y \geq r_i/e_i$*, with* $v_i < 2\,\delta/d_Y$ *and* $\sum_{i=1}^{\lambda} e_i\, f_i \geq \frac{d_Y}{2}$*. Not taking into account the cost of univariate factorisations, it takes an expected* $\mathcal{O}(\mathsf{M}(d_Y\, \delta) \log(d_Y)) \subset \mathcal{O}\tilde{\ }(d_Y\, \delta)$ *arithmetic operations over* $\mathbb{K}$.

Remark 5. The key idea is to use tighter truncation bounds than in [31, 33]. Proposition 5 says that $n \in \mathcal{O}(\delta/d_Y)$ is enough to get some informations (at least half of the singular parts of Puiseux series). This requires a slight modification of [33, Algorithm `ARNP`]: n is updated in a different way. When there is a transform $X \leftarrow X^q$, it must be multiplied by q; also, it cannot be divided by the degree t of the found extension anymore. These points are actually compensated by algorithm `WPT`, which divides the degree in Y by the same amount (it eliminates all the conjugates). The size of the input polynomial H is thus bounded by $\mathcal{O}(\delta)$ elements of $\mathbb{K}$ (cf Section 3.4).

3.3 Using tight truncations bounds.

By a carefull study of the `RNPuiseux` algorithm, we get an optimal truncation bound to compute a RPE of a monic polynomial F with this algorithm or `Half-RNP`. From this study, we also deduce an exact relation between δ and this optimal bound. In this section, for $1 \le i \le \rho$, we let $m_{i,h}\, a + q_{i,h}\, b = l_{i,h}$, $1 \le h \le g_i$ be the successive edges encountered during the computation of the expansion R_i with `RNPuiseux`, and denote

$$N_i := \sum_{h=1}^{g_i} \frac{l_{i,h}}{q_{i,1}\cdots q_{i,h}}.$$

Lemma 5. *For any $1 \le i \le \rho$, we have $N_i = \frac{r_i}{e_i} + v_i$.*

Proof. Denote $R_i = (\gamma_i\, X^{e_i}, \Gamma_i(X,Y))$ with $\Gamma_i(X,Y) = \Gamma_{i,0}(X) + X^{r_i}\, Y$. By the definition of the Puiseux transformations, we have $(0,1) \in \mathcal{N}(G_i)$ for

$$G_i(X,Y) := \frac{F(\gamma_i\, X^{e_i}, \Gamma_i(X,Y))}{X^{N_i\, e_i}},$$

i.e. $\upsilon_X\,(\partial_Y G_i(X,0)) = 0$. This is $\upsilon_X(X^{r_i}\, F_Y(\gamma_i\, X^{e_i}, \Gamma_{i,0}(X))) = N_i\, e_i$, or:

$$N_i = \frac{r_i + \upsilon_X(F_Y(\gamma_i\, X^{e_i}, \Gamma_{i,0}(X)))}{e_i} = \frac{r_i}{e_i} + \upsilon_X(F_Y(X, \Gamma_{i,0}((X/\gamma_i)^{1/e_i}))) = \frac{r_i}{e_i} + v_i. \quad \square$$

Remark 6. This result shows that N_i does not depend on the algorithm. Nevertheless, the proof above relies on algorithm `RNPuiseux` because it computes *precisely* the singular part of all Puiseux series thanks to the *modified* Newton polygon [33, Definition 6]. The algorithm `Half-RNP` introduces two differences:

- The Abhyankar's trick does not change the value of the N_i. After applying it, the next value $\frac{l}{q}$ is just the addition of the $\frac{l_i}{q_i}$ we would have found with `RNPuiseux` (the concerned slopes being the sequence of integer slopes that compute common terms for *all* Puiseux series, plus the next one). See Example 4 below.
- Not using the modified Newton polygon $\mathcal{N}^\star$ can only change the last value $\frac{l}{q}$ (when the coefficient of $X^{r/e}$ is 0). This has no impact on the proof of Lemma 6 below.

In the remaining of this paper, we will define N_i as $\frac{r_i}{e_i} + v_i$.

Example 4. Let's assume that F is an irreducible polynomial with Puiseux series $S(X) = X^{1/2} + X + X^{3/2} + X^2 + X^{9/4}$. The successive values for (l,q) are:

- $(4,2)$, $(2,1)$, $(2,1)$, $(2,1)$ and $(2,2)$ with the `RNPuiseux` algorithm. We thus get $N = 2 + 1 + 1 + 1 + \frac{1}{2} = \frac{11}{2}$.
- $(4,2)$, $(14,2)$ with the `Half-RNP` algorithm (assuming high enough truncation). We thus get $N = 2 + \frac{7}{2} = \frac{11}{2}$.

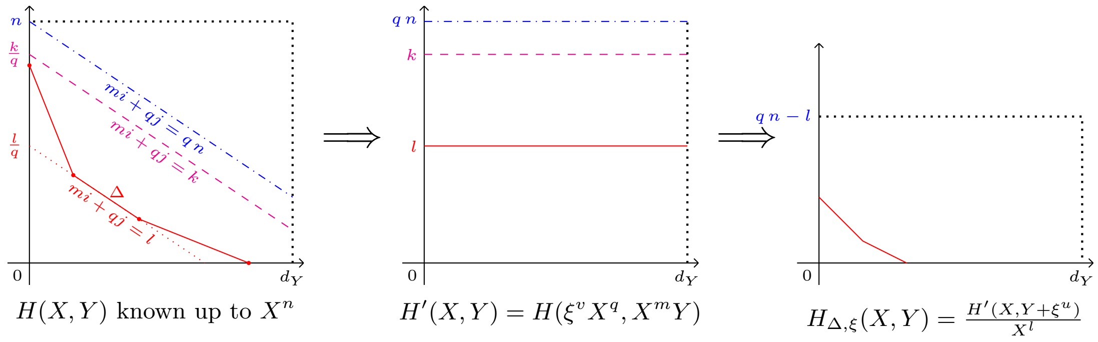


Figure 2: Change of variables for a Puiseux transform

Lemma 6. *Let $n_0 \in \mathbb{N}$. To compute the RPE R_i with certified precision $n_0 \geq \frac{r_i}{e_i}$, it is necessary and sufficient to run* `Half-RNP` *with truncation bound $n = n_0 + v_i$. In particular, to ensure the computation of the singular part of R_i, it is necessary and sufficient to use a truncation bound $n \geq N_i$.*

Proof. First note that starting from H known up to X^n, the greatest n_1 so that we can certify $H_{\Delta,\xi} := H(\xi^v X^q, X^m (Y+\xi^u))/X^l$ up to X^{n_1} is precisely $n_1 = q\,n - l$ (see Figure 2; details are in [31, Proof of Lemma 2]). This explains the truncation update of line 10.

We now distinguish two cases, according to whether the coefficient in $X^{\frac{r_i}{e_i}}$ of any Puiseux series associated to R_i is zero or not. If not, then starting from a truncation bound $n = n' + N_i$, we get $n_1 = q\,n' + q\,N_i - l$. By construction, $q\,N_i - l$ is precisely the "N_i" of the associated RPE of $H_{\Delta,\xi}$. By induction, we finish at the last call of the algorithm associated to the RPE R_i with a truncation bound $n = e_i\,n'$. Moreover, we have $\deg_Y(H) = 1$ and $\pi = (\gamma_i X^{e_i}, \Gamma_i(X) + \alpha_i X^{r_i} Y)$. Hence, the ouptut R_i is known with precision $n' + \frac{r_i}{e_i}$. We conclude thanks to Lemma 5 by taking $n' = n_0 - \frac{r_i}{e_i}$.

Finally, if the coefficient in $X^{\frac{r_i}{e_i}}$ of any Puiseux series associated to R_i is zero, we will have $\pi = (\gamma_i X^{e_i}, \Gamma_i(X) + \alpha_i X^{\eta_i} Y)$ with $\eta_i > r_i$. If this is the case, then that means that the at the previous step, we already computed some zero coefficients, thus losing the same precision $\eta_i - r_i$. This does not change the result. □

This proves that N_i is an optimal bound to compute the singular part of the RPE R_i. We now bound it.

Lemma 7. *We have $\frac{r_i}{e_i} \leq v_i$.*

Proof. This is written in the proof of [31, Proposition 5, page 204]. □

Corollary 4. *We have $v_i \leq N_i \leq 2\,v_i$.*

Proof. Straightforward consequence of Lemmas 5 and 7. □

We finally deduce global bounds:

Proposition 6. *At least $\frac{d_Y}{2}$ Puiseux series $S_{i,j,k}$ satisfy $v_i < 2\,\delta/d_Y$ and $N_i < 4\,\delta/d_Y$.*

Proof. Assume the R_i ordered s.t. $v_i \leq v_{i+1}$, and let λ s.t. $\sum_{i=1}^{\lambda-1} e_i\, f_i < \frac{d_Y}{2} \leq \sum_{i=1}^{\lambda} e_i\, f_i$ (i.e. $\sum_{i=\lambda+1}^{\rho} e_i\, f_i \leq \frac{d_Y}{2} < \sum_{i=\lambda}^{\rho} e_i\, f_i$ by Proposition 1). Then we have

$$\delta = \sum_{i=1}^{\rho} v_i\, e_i\, f_i \geq \sum_{i=\lambda}^{\rho} v_i\, e_i\, f_i \geq v_\lambda \sum_{i=\lambda}^{\rho} e_i\, f_i > v_\lambda \frac{d_Y}{2},$$

the first equality being a resultant property (see e.g. [18, Exercise 6.12]). Hence, for all $i \leq \lambda$, we have $v_i \leq v_\lambda < 2\delta/d_Y$, thus $N_i < 4\delta/d_Y$ by Corollary 4. The claim follows. □

3.4 Complexity results and proof of Theorem 4.

Proposition 7. *Not taking into account the cost of univariate factorisations, running* `Half-RNP`$(F, Z, n, (X, Y))$ *takes an expected $\mathcal{O}(\mathsf{M}(n\, {d_Y}^2) \log(d_Y))$ operations over $\mathbb{K}$.*

Proof. Let's consider a function call to `Half-RNP`(H, P, n_H, π) and denote $d_P = \deg_Z(P)$. We distinguish two kind of lines (for both, note the bound $n\, d_Y \geq n_H \deg_Y(H)\, d_P$):

(Type 1) By Lemma 4, Line 2 takes less than $\mathcal{O}(\mathsf{M}(n\, d_Y))$ operations over $\mathbb{K}$. So do Lines 1 and 9, by respectively Lemmas 4 and 2, using Remark 4 and $e_i\, f_i \leq d_Y$.

(Type 2) Lines 10 and 11 are $\mathcal{O}(\mathsf{M}(q\, d_\phi\, n\, d_Y))$ from respectively Lemma 2 and Proposition 3. By Proposition 4, so is Line 7, while Line 8 costs $\mathcal{O}((d_P d_\phi)^{\frac{\omega+1}{2}})$.

From Lemma 3, when $q = d_\phi = 1$, we must have a branch separation. Therefore, this happens at most $\rho - 1$ times (more precisely, the number of pairs (Δ, ϕ) with $q = d_\phi = 1$ while considering all recursive calls is bounded by ρ). This means that the sum of the costs for these cases is less than $\mathcal{O}(\rho\, \mathsf{M}(n\, d_Y)) \subset \mathcal{O}(\mathsf{M}(n\, {d_Y}^2))$.

To conclude the proof, we still have to deal with all the cases where $q > 1$ or $d_\phi > 1$. In such a case, Type 2 lines are the costly ones. Moreover, we can bound q by e_i and $d_P d_\phi$ by f_i for any RPE R_i issued from (Δ, ϕ). But for each RPE R_i, such situation cannot happen more than $\log(e_i\, f_i) \leq \log(d_Y)$ times (before and/or after separation of this branch with other ones). From Definition 9, that means we can bound the total cost for all these cases by $\mathcal{O}((\mathsf{M}(\sum_{i=1}^{\rho} e_i\, f_i\, n\, d_Y) + \sum_{i=1}^{\rho} f_i^{\frac{\omega+1}{2}}) \log(d_Y)) \subset \mathcal{O}(\mathsf{M}(n\, {d_Y}^2) \log(d_Y))$. □

Proof of Proposition 5. As far as correctness is concerned, we only have to take care of truncations and the precision of the output: other points are considered in previous papers of the first author [29, 31, 33] (note also [14, Section 4.1] concerning the construction of the output). From Lemma 6, a function call `Half-RNP`$(F, Z, 6\delta/d_Y, (X, Y))$ provides (at least) the Puiseux series satisfying $v_i < 2\delta/d_Y$ with precision $4\delta/d_Y$ or greater. As $r_i/e_i \leq v_i$ from Lemma 7, their singular parts are known. Also, from Proposition 6, we get at least half of the Puiseux series of F. Complexity is Proposition 7. □

4 A divide and conquer algorithm.

We keep notations of Sections 1 and 3, and prove in this section the following result:

Theorem 5. *Not taking into account the cost of univariate factorisations, there exists an algorithm that computes the singular part of all rational Puiseux expansions of F above $x_0 = 0$ in less than $\mathcal{O}(\mathsf{M}(d_Y\,\delta)\,\log(d_Y\,\delta) + \mathsf{M}(d_Y)\,\log^2(d_Y))$ arithmetic operations.*

Assuming that F is monic, our strategy can be summarised as follows:

1. Run `Half-RNP`$(F, Z, 6\,\delta/d_Y, (X, Y))$. If this provides all RPEs of F, we are done. If not, from Section 3, we get at least half of the Puiseux series of F, satisfying $v_i < 2\,\delta/d_Y$, and known with precision $4\,\delta/d_Y$ or more.
2. From these Puiseux series, construct the associated irreducible factors and their product G with precision $4\,\delta/d_Y$; cf Section 4.1. Note that $\deg_Y(G) \geq d_Y/2$.
3. Compute its cofactor H by euclidean division modulo $X^{4\,\delta/d_Y+1}$.
4. Compute the Bézout relation $U\,G + V\,H = X^\kappa \mod X^{\kappa+1}$ via [24, Algorithm 1]. We prove in Section 4.2 that $\kappa \leq 2\,\delta/d_Y$.
5. Using this relation, lift the factorisation $F = G\,H \mod X^{4\,\delta/d_Y+1}$ to precision δ using a variant of the Hensel lemma. See Section 4.3.
6. Finally, apply the main algorithm recursively on H; as the degree in Y is at least divided by two each time, this is done at most $\log(d_Y)$ times, for a total cost only multiplied by 2. This is detailed in Section 4.4.

If F is not monic (this assumption is not part of Theorem 5), first use Hensel lifting to compute the factor F_∞ corresponding to RPEs centered at $(0, \infty)$ up to precision X^δ. Then, compute the RPEs of F_∞ as "inverse" of the RPEs of its reciprocal polynomial (which is monic by construction). Details are provided in Section 4.5.

4.1 Computing the norm of a RPE.

Lemma 8. *Let $R_1, \cdots, R_\lambda$ be a set of $\mathbb{K}$-RPEs not centered at $(0, \infty)$. For each R_i, we denote $(S_{ijk})_{jk}$ its associated Puiseux series. Let*

$$\nu = \max_{1\leq i\leq\lambda} \sum_{\substack{(i',j',k')\\ \neq(i,j,k)}} v_X(S_{ijk}(X) - S_{i'j'k'}(X))$$

and assume that the R_i are known with precision $n \geq \nu$. Then there exists an algorithm `NormRPE` *that computes $G \in \mathbb{K}[X, Y]$ monic with $\deg_Y(G) = \sum_{i=1}^{\lambda} e_i\,f_i$, $\deg_X(G) = n+\nu$, and such that the RPE of G with precision n are precisely the R_i. It takes less than $\mathcal{O}(\mathsf{M}(n\,\deg_Y(G)^2)\,\log(n\,\deg_Y(G))) \subset \tilde{\mathcal{O}}(n\,\deg_Y(G)^2)$ arithmetic operations over $\mathbb{K}$.*

Proof. Denote $P_i \in \mathbb{K}[Z]$ so that $R_i = (\gamma_i(Z)\, T^{e_i}, \Gamma_i(Z,T))$ is defined over $\mathbb{K}_{P_i}$. Compute

$$A_i = \prod_{j=0}^{e_i-1} \left(Y - \Gamma_i \left(Z, \zeta_{e_i}^j \left(\frac{X}{\gamma_i} \right)^{\frac{1}{e_i}} \right) \right) \mod (X^{n+\nu+1}, P_i(Z))$$

for $1 \le i \le \lambda$. As $n \ge \nu$, it takes $\mathcal{O}(\mathsf{M}(e_i^2\, n\, f_i) \log(e_i))$ operations in $\mathbb{K}$ using a sub-product tree. Then, compute $G_i = \mathrm{Res}_Z(A_i, P_i) \mod X^{n+\nu+1}$. Adapting [18, Corollary 11.21, page 332] to a polynomial with three variables, this is $\mathcal{O}(f_i\, \mathsf{M}(n\, e_i\, f_i) \log(n\, e_i\, f_i))$. Summing over i these two operations, this fits into our bound. Finally, compute G the product of the G_i modulo $X^{n+\nu+1}$ in less than $\mathcal{O}(\mathsf{M}(n \deg_Y(G)) \log(\deg_Y(G)))$ using a sub-product tree [18, Algorithm 10.3, page 297]. It has the required properties. □

4.2 Lifting order.

Our algorithm requires to lift some analytic factors G, H of F which are not coprime modulo (X). To this aim, we will generalise the classical Hensel lifting. The first step is to compute a generalized Bézout relation $UG + VH = X^\kappa$ with $\kappa \in \mathbb{N}$ minimal.

Definition 11. Let $G, H \in \mathbb{K}[[X]][Y]$ coprime. The *lifting order* of G and H is:

$$\kappa(G,H) := \inf\{k \in \mathbb{N},\ X^k \in (G,H)\}.$$

We now provide an upper bound for the lifting order that is sufficient for our purpose.

Proposition 8. *If $F = G \cdot H$ with H monic, we have $\kappa(G,H) \le \max_{H(S)=0} v_X(F_Y(S))$.*

Proof. Let $U\,G + V\,H = X^\kappa$ in $\mathbb{K}[[X]][Y]$, with $\kappa = \kappa(G,H)$ minimal. Up to perform the euclidean division of U by H, we may assume $\deg_Y(U) < \deg_Y(H) =: d$. Moreover, minimality of κ and monicity of H impose $v_X(U) = 0$. Denoting $S_1, \cdots, S_d$ the Puiseux series of H, we have $U(S_i)\, G(S_i) = X^\kappa$ for $1 \le i \le d$. Using interpolation, we get

$$U = \sum_{i=1}^{d} \frac{X^\kappa}{G(S_i)\, H_Y(S_i)} \prod_{j \ne i} (Y - S_j) = \sum_{i=1}^{d} \frac{X^\kappa}{F_Y(S_i)} \prod_{j \ne i} (Y - S_j).$$

As $v_X(U) = 0$ and $v_X(S_j) \ge 0$ (H is monic), we have $\kappa \le \max_{1 \le i \le d} v_X(F_Y(S_i))$. □

Corollary 5. *Assume that $F \in \mathbb{K}[[X]][Y]$ is a non irreducible monic polynomial. Then there exists a factorisation $F = G\,H$ in $\mathbb{K}[[X]][Y]$ such that $\kappa(G,H) \le 2\,\delta/d_Y$.*

Proof. From Proposition 6, there exist $\lambda \ge 1$ RPE $R_1, \cdots, R_\lambda$ of F such that $v_i < 2\,\delta/d_Y$ for all $i \le \lambda$. Considering $H = \prod_{i=1}^{\lambda} F_i$ and $G = \prod_{i=\lambda+1}^{\rho} F_i$ (with F_i the analytic factor associated to R_i - see Section 2.1), we are done from Proposition 8. □

The relation $U\,G + V\,H = X^\kappa \mod X^{\kappa+1}$ can be computed in $\mathcal{O}(\mathsf{M}(d_Y\, \kappa) \log(\kappa) + \mathsf{M}(d_Y)\, \kappa \log(d_Y))$ [24, Corollary 1]. This is $\mathcal{O}(\mathsf{M}(\delta) \log(\delta))$ for (G,H) of Corollary 5

4.3 Adaptation of Hensel's lemma to our context.

We generalise the classical Hensel lemma [18, section 15.4] when polynomials are not coprime modulo X. First, the following algorithm "double the precision" of the lifting: given F, G, H, U, $V \in \mathbb{K}[X, Y]$ with H monic in Y, and $n_0, \kappa \in \mathbb{N}$ satisfying

- $F = G\,H \mod X^{n_0}$ with $n_0 > 2\,\kappa$,
- $U\,G + V\,H = X^{\kappa} \mod X^{n_0-\kappa}$ with $\deg_Y(U) < \deg_Y(H)$, $\deg_Y(V) < \deg_Y(G)$,

it outputs polynomials $\tilde{G}$, $\tilde{H}$, $\tilde{U}$, $\tilde{V} \in \mathbb{K}[X, Y]$ with $\tilde{H}$ monic in Y such that:

- $F = \tilde{G}\,\tilde{H} \mod X^{2\,(n_0-\kappa)}$, with $\tilde{G} = G \mod X^{n_0-\kappa}$ and $\tilde{H} = H \mod X^{n_0-\kappa}$,
- $\tilde{U}\,\tilde{G} + \tilde{V}\,\tilde{H} = X^{\kappa} \mod X^{2\,n_0-3\,\kappa}$; $\deg_Y(\tilde{V}) < \deg_Y(\tilde{G})$, $\deg_Y(\tilde{U}) < \deg_Y(\tilde{H})$.

In what follows, `QuoRem` denotes the classical euclidean division algorithm.

Algorithm: `HenselStep`$(F, G, H, U, V, n_0, \kappa)$

$\alpha \leftarrow X^{-\kappa}(F - G \cdot H) \mod X^{2\,(n_0-\kappa)}$;
$Q, R \leftarrow \mathtt{QuoRem}_Y(U \cdot \alpha, H) \mod X^{2\,(n_0-\kappa)}$;
$\tilde{G} \leftarrow G + \alpha \cdot V + Q \cdot G \mod X^{2\,(n_0-\kappa)}$;
$\tilde{H} \leftarrow H + R \mod X^{2\,(n_0-\kappa)}$;
$\beta \leftarrow X^{-\kappa}(U \cdot \tilde{G} + V \cdot \tilde{H}) - 1 \mod X^{2\,n_0-3\,\kappa}$;
$S, T \leftarrow \mathtt{QuoRem}_Y(U \cdot \beta, \tilde{H}) \mod X^{2\,(n_0-\kappa)}$;
$\tilde{U} \leftarrow U - T \mod X^{2\,n_0-3\,\kappa}$;
$\tilde{V} \leftarrow V - \beta \cdot V - S \cdot \tilde{G} \mod X^{2\,n_0-3\,\kappa}$;
return $\tilde{H}$, $\tilde{G}$, $\tilde{U}$, $\tilde{V}$

Lemma 9. *Algorithm* `HenselStep` *is correct; it runs in* $\mathcal{O}(\mathsf{M}(n_0\,d_Y))$ *operations in* $\mathbb{K}$.

Proof. From $\alpha \equiv 0 \mod X^{n_0-\kappa}$ (thus $Q \equiv 0 \mod X^{n_0-\kappa}$ and $R \equiv 0 \mod X^{n_0-\kappa}$ from [18, Lemma 15.9, (ii), page 445]) and $U \cdot G + V \cdot H - X^{\kappa} \equiv 0 \mod X^{n_0-\kappa}$, we have $\tilde{G} \equiv G \mod X^{n_0-\kappa}$, $\tilde{H} \equiv H \mod X^{n_0-\kappa}$ and

$$\begin{aligned} F - \tilde{G} \cdot \tilde{H} &\equiv F - (G + \alpha \cdot V + Q \cdot G) \cdot (H + \alpha \cdot U - Q \cdot H) \\ &\equiv \alpha(X^{\kappa} - V \cdot H - U \cdot G) - \alpha^2 \cdot U \cdot V - Q \cdot \alpha(U \cdot G - V \cdot H) + Q^2 \cdot G \cdot H \\ &\equiv 0 \mod X^{2\,(n_0-\kappa)}. \end{aligned}$$

From $\beta \equiv 0 \mod X^{n_0-2\,\kappa}$ and $U \cdot \tilde{G} + V \cdot \tilde{H} - X^{\kappa} \equiv 0 \mod X^{n_0-\kappa}$, we have:

$$\begin{aligned} \tilde{U} \cdot \tilde{G} + \tilde{V} \cdot \tilde{H} - X^{\kappa} &\equiv (U - U \cdot \beta + S \cdot \tilde{H}) \cdot \tilde{G} + (V - \beta \cdot V - S \cdot \tilde{G}) \cdot \tilde{H} - X^{\kappa} \\ &\equiv U \cdot \tilde{G} + V \cdot \tilde{H} - X^{\kappa} - \beta \cdot (U \cdot \tilde{G} + V \cdot \tilde{H}) \\ &\equiv \beta \cdot (X^{\kappa} - U \cdot \tilde{G} - V \cdot \tilde{H}) \equiv 0 \mod X^{2\,n_0-3\,\kappa}. \end{aligned}$$

Conditions on the degrees in Y for $\tilde{H}$ and $\tilde{U}$ are obvious (thus is the monicity of $\tilde{H}$). The complexity result is similar to [18, Theorem 9.6, page 261]. $\square$

Assuming we start from a relation $F = G\,H \mod X^{2\,\kappa+1}$ with a Bézout relation $U\,G + V\,H = X^\kappa \mod X^{\kappa+1}$, we thus can iterate this algorithm up to the wanted precision:

Lemma 10. *Given F, G, H as in the input of algorithm* `HenselStep` *with $n_0 = 2\,\kappa + 1$, there exists an algorithm* `Hensel` *that computes polynomials $(\tilde{G}, \tilde{H})$ as in the output of* `HenselStep` *for any precision $n \in \mathbb{N}$, additionally satisfying:*

- $\tilde{G} = G \mod X^{\kappa+1}$, $\tilde{H} = H \mod X^{\kappa+1}$ *and* $F = \tilde{G} \cdot \tilde{H} \mod X^{n+2\,\kappa}$*;*
- *if there are $G^\star$, $H^\star \in \mathbb{K}[X,Y]$ satisfying $F = G^\star \cdot H^\star \mod X^{n+2\,\kappa}$, then $\tilde{G} = G^\star \mod X^n$ and $\tilde{H} = H^\star \mod X^n$.*

It takes less than $\mathcal{O}(\mathsf{M}(n\,d_Y) + \mathsf{M}(\kappa\,d_Y)\,\log(\kappa\,d_Y))$ operations in $\mathbb{K}$.

Proof. The algorithm runs as follows:

1. Compute $U, V \in \mathbb{K}[X,Y]$ s.t. $U \cdot G + V \cdot H = X^\kappa \mod X^{\kappa+1}$ [24, Algorithm 1].
2. Double the value $n_0 - 2\,\kappa$ at each call of `HenselStep`, until $n_0 - 2\,\kappa \geq n + \kappa$.

Correctness and complexity follow Lemma 9 (using [24, Corollary 1] for the computation of U and V). Finally, uniqueness of the result is an adaptation of [18, Theorem 15.14, page 448] (this works because we take a precision satisfying $n_0 - 2\,\kappa \geq n + \kappa$). ☐

Remark 7. Note that if $G(0,Y)$ and $H(0,Y)$ are coprime, then $\kappa = 0$ and this result is the classical Hensel lemma.

4.4 The divide and conquer algorithm for monic polynomials.

We provide our divide and conquer algorithm. Algorithm `Quo` outputs the quotient of the euclidean division in $\mathbb{K}[[X]][Y]$ modulo a power of X, and $\#\mathcal{R}$ is the cardinal of $\mathcal{R}$.

Algorithm: `MonicRNP`(F, n)

In: $F \in \mathbb{K}[X,Y]$, separable and monic in Y ; $n \in \mathbb{N}$ "big enough".

Out: the singular part (at least) of all the RPEs of F above $x_0 = 0$.

if $d_Y < 6$ **then return** `Half-RNP`$(F, Z, n, (X,Y))$ **else** $\eta \leftarrow 6\,n/d_Y$;
$\mathcal{R} \leftarrow$ `Half-RNP`$(F, Z, \eta, (X,Y))$;
Keep in $\mathcal{R}$ the RPEs with $v_i < \eta/3$; `// known with precision` $\geq 2\eta/3$
if $\#\mathcal{R} = d_Y$ **then return** $\mathcal{R}$;
$G \leftarrow$ `NormRPE`$(\mathcal{R}, 2\eta/3)$;
$H \leftarrow$ `Quo`$(F, G, 2\eta/3)$;
$G, H \leftarrow$ `Hensel`(F, G, H, n);
return $\mathcal{R} \cup$ `MonicRNP`(H, n);

Proposition 9. *If $n \geq \delta$,* `MonicRNP`(F, n) *returns the correct ouput in an expected $\mathcal{O}(\mathsf{M}(d_Y\,n)\log(d_Y\,n))$ operations in $\mathbb{K}$, plus the cost of univariate factorisations.*

Proof. We start with correctness. As precision $n \geq \delta$ is sufficient to compute the singular parts of all Puiseux series via algorithm `Half-RNP`, the output is correct when $d_Y < 6$. When $d_Y \geq 6$, Line 2 provides a set of RPEs $(R_i)_{1 \leq i \leq \lambda}$ known with precision $\eta - v_i$ by Lemma 6. At line 5, we keep in $\mathcal{R}$ the RPEs R_i such that $v_i < \eta/3$; they are thus known with precision at least $2\,\eta/3$. Also, we have $\deg_Y(G) \geq d_Y/2 \geq \deg_Y(H)$ from Proposition 6. Finally, input of the `Hensel` algorithm is correct since $\kappa(G, H)$ is less than $\eta/3$ by Proposition 8 and we know the factorisation $F = G \cdot H \mod X^{2\,\eta/3+1}$.

We now focus on complexity. By Proposition 7, Lines 1 (d_Y is constant) and 2 are respectively $O(\mathsf{M}(n))$ and $\mathcal{O}(\mathsf{M}(n\,d_Y)\,\log(d_Y))$. Lines 5, 6 and 7 take respectively $\mathcal{O}(\mathsf{M}(n\,d_Y)\,\log(n\,d_Y))$, $\mathcal{O}(\mathsf{M}(n\,d_Y))$ and $\mathcal{O}(\mathsf{M}(n\,d_Y) + \mathsf{M}(\delta)\,\log(\delta))$ by respectively Lemma 8, division via Newton iteration [18, Theorem 9.4] and Lemma 10. This fits into our result (remember $n \geq \delta$). Finally, as $\deg_Y(H) \leq d_Y/2$, we conclude from Lemma 1. $\square$

4.5 Dealing with the non monic case: proof of Theorem 5.

Proposition 10. *There exists an algorithm* `Monic` *that given $n \in \mathbb{N}$ and $F \in \mathbb{K}[X, Y]$ primitive in Y, returns $u \in \mathbb{K}[X]$ and $F_0, F_\infty \in \mathbb{K}[X, Y]$ s.t. $F = u\,F_0 F_\infty \mod X^n$, with F_0 monic in Y, $F_\infty(0, Y) = 1$, and $u(0) \neq 0$ with $\mathcal{O}(\mathsf{M}(n\,d_Y))$ operations over $\mathbb{K}$.*

Proof. This is [25, Algorithm Q, page 33] (see the proof of Proposition 3). $\square$

We can now give our main algorithm `RNP`. It computes the singular part of all RPEs of F above $x_0 = 0$, including those centered at $(0, \infty)$. This algorithm, called with parameters (F, δ) is the algorithm mentioned in Theorem 5.

Algorithm: `RNP`(F, n)
In: $F \in \mathbb{K}[X, Y]$, separable in Y and $n \in \mathbb{N}$ "big enough".
Out: the singular part (at least) of all the RPEs of F above $x_0 = 0$

$(u, F_0, F_\infty) \leftarrow$ `Monic`(F, n);
$\tilde{F}_\infty \leftarrow Y^{\deg_Y(F_\infty)} F_\infty(X, 1/Y)$;
$\mathcal{R}_\infty \leftarrow$ `MonicRNP`$(\tilde{F}_\infty, n)$;
Inverse the second element of each $R \in \mathcal{R}_\infty$;
return `MonicRNP`$(F_0, n) \cup \mathcal{R}_\infty$;

The proof of Theorem 5 follows immediately from the following proposition:

Proposition 11. *Not taking into acount the cost of univariate factorisations,* `RNP`(F, δ) *returns the correct output with an expected $\mathcal{O}(\mathsf{M}(d_Y\,\delta)\,\log(d_Y\,\delta))$ arithmetic operations.*

There is one delicate point in the proof of Proposition 11: we need to invert the RPEs of $\tilde{F}_\infty$ and it is not clear that the truncation bound $n = \delta$ is sufficient for recovering in such a way the singular part of the RPEs of F_∞ (see also Remark 8 below). We will need the two following results:

Lemma 11. *Consider two distinct Puiseux series S and S_0. Then we have*

$$\upsilon_X\left(\frac{1}{S}-\frac{1}{S_0}\right)=\upsilon_X(S-S_0)-\upsilon_X(S)-\upsilon_X(S_0).$$

Proof. If $\upsilon_X(S)\neq\upsilon_X(S_0)$, one can assume $\upsilon_X(S)<\upsilon_X(S_0)$, i.e. $\upsilon_X(S-S_0)=\upsilon_X(S)$ and $\upsilon_X\left(\frac{1}{S}-\frac{1}{S_0}\right)=\upsilon_X(S_0)$. If $\upsilon_X(S)=\upsilon_X(S_0)=\alpha$, then $\frac{X^\alpha}{S} \mod X^n$ is uniquely determined from $X^{-\alpha}S \mod X^n$ (same for S_0). If $s=\upsilon_X(S-S_0)-\alpha$, we have $X^{-\alpha}S=X^{-\alpha}S_0 \mod X^s$, i.e. $\frac{X^\alpha}{S}=\frac{X^\alpha}{S_0} \mod X^s$ and $\alpha+\upsilon_X\left(\frac{1}{S}-\frac{1}{S_0}\right)\geq s$. Similarly, denoting $s_0=\alpha+\upsilon_X\left(\frac{1}{S}-\frac{1}{S_0}\right)$, we get $\upsilon_X(S-S_0)-\alpha\geq s_0$, concluding the proof. □

Proposition 12. *Let $F_\infty\in\mathbb{K}[X,Y]$ with $F_\infty(0,Y)=1$ and denote $\tilde{F}_\infty$ its reciprocal polynomial according to Y. For each RPE $R_i=(\lambda_i X^{e_i},\Gamma_i)$ of F_∞, denote $s_i:=\upsilon_X(\Gamma_i)$ (so $s_i<0$), r_i its regularity index and $\tilde{R}_i$ the associated RPE of $\tilde{F}_\infty$. The function call* `MonicRNP`$(\tilde{F}_\infty,\delta_{F_\infty})$ *computes each RPE $\tilde{R}_i$ with precision at least $\frac{r_i-2\,s_i}{e_i}$.*

Proof. Denote $d=\deg_Y(F_\infty)$, $v=\upsilon_X(\mathrm{lc}_Y(F_\infty)$, $S_1,\cdots,S_d$ the Puiseux series of F_∞ and S_{k_i} one of them associated to the RPE R_i of F_∞ we are considering. Then $\frac{s_i}{e_i}=\upsilon_X(S_{k_i})$ and $\upsilon_X(S_{k_i}-S_j)\leq\frac{r_i}{e_i}$ for $j\neq k_i$ by definition of r_i. Let i_0 satisfying $\upsilon_X(S_{k_i}-S_{i_0})=\max_{j\neq k_i}\upsilon_X(S_{k_i}-S_j)$ (several values of i_0 are possible). We distinguish three cases:

1. $\upsilon_X(S_{k_i})=\upsilon_X(S_{i_0})$; then either $\upsilon_X(S_{k_i}-S_{i_0})=\frac{r_i}{e_i}$, or $e_{i_0}=q\,e_i$ with $q>1$. In the latter case, there exist q conjugates Puiseux series $S_{i_0}^{[0]},\cdots,S_{i_0}^{[q-1]}$ of S_{i_0} such that $\upsilon_X(S_{k_i}-S_{i_0})=\upsilon_X\left(S_{k_i}-S_{i_0}^{[j]}\right)$, thus $\sum_{j=0}^{q-1}\upsilon_X\left(S_{k_i}-S_{i_0}^{[j]}\right)\geq\frac{r_i}{e_i}$; see [31, Case 3 in Proof of Proposition 5, pages 204 and 205] for details.
2. $\upsilon_X(S_{k_i})>\upsilon_X(S_{i_0})$. Then $r_i=s_i$ and $\upsilon_X(S_{k_i})>\upsilon_X(S_j)$ for $j\neq k_i$ by definition of i_0. This means $\frac{v}{d}\geq-\frac{s_i}{e_i}$ as $v=\sum_{k=1}^d-\frac{s_k}{e_k}$ from [31, Lemma 1, page 198].
3. $\upsilon_X(S_{k_i})<\upsilon_X(S_{i_0})$. Then $\upsilon_X(S_{k_i}-S_{i_0})=s_i=r_i$. We can also assume that $\upsilon_X(S_j)\neq\upsilon_X(S_{k_i})$ for all $j\neq k_i$: if $\upsilon_X(S_j)=\upsilon_X(S_{k_i})$, then $\upsilon_X(S_{k_i}-S_j)=\upsilon_X(S_{k_i}-S_{i_0})$ and one could use $i_0=j$ and deal with it as Case 1.

We can now prove Proposition 12. First, for Case 1, knowing $\frac{1}{S_{k_i}}$ with precision $\tilde{v}_i:=\upsilon_X\left(\partial_Y\tilde{F}_\infty\left(\frac{1}{S_{k_i}}\right)\right)$ is sufficient: from Lemma 11, we have $\tilde{v}_i=\sum_{j\neq i}\upsilon_X\left(\frac{1}{S_{k_i}}-\frac{1}{S_j}\right)=\sum_{j\neq i}\upsilon_X\left(S_{k_i}-S_j\right)-\upsilon_X(S_{k_i})-\upsilon_X(S_j)$. As either $\upsilon_X(S_{k_i}-S_{i_0})-\upsilon_X(S_{k_i})-\upsilon_X(S_{i_0})=\frac{r_i-2\,s_i}{e_i}$ or $\sum_{j=0}^{q-1}\upsilon_X\left(S_{k_i}-S_{i_0}^{[j]}\right)-\upsilon_X(S_{k_i})-\upsilon_X(S_{i_0}^{[j]})\geq\frac{r_i-2\,s_i}{e_i}$, we are done.

Then, concerning Case 2, from Proposition 9 and Lemma 6, we know the RPE R_i with precision at least $v_i+\frac{v}{d}$, that is at least $\frac{v}{d}\geq-\frac{s_i}{e_i}$. As $r_i=s_i$, this is at least $\frac{r_i-2\,s_i}{e_i}$.

Finally, Case 3 requires more attention. Let's first assume that $\upsilon_X(S_{k_i})>\upsilon_X(S_j)$ for some $j\neq i$; then $\upsilon_X(S_{k_i}-S_j)-\upsilon_X(S_{k_i})-\upsilon_X(S_j)=\upsilon_X(S_{k_i})=-\frac{s_i}{e_i}$, and we are done since $r_i=s_i$. If not, then we have $\upsilon_X(S_{k_i})<\upsilon_X(S_j)$ for all j. This means that $e_i=f_i=1$, and that $\mathcal{N}(\tilde{F}_\infty)$ has an edge $[(0,v),(1,v-s_i)]$, which is associated to $\tilde{R}_i$. It is enough to prove that the truncation bound used when dealing with this Puiseux series is at least v. As long as this is not the case, this edge is not considered from the definition of $\mathcal{N}_n(H)$; also, at each recursive call of `MonicRNP` (Line 8), the value of the truncation bound η increases (since the degree in Y is at least divided by 2). In the worst case, we end with a degree 1 polynomial, thus using $\eta = \delta_{F_\infty} \geq v$. This concludes. □

Proof of Proposition 11. Let us show that the truncation bound for $\tilde{F}_\infty$ is sufficient for recovering the singular part of the Puiseux series of F_∞. First note that the inversion of the second element is done as follows: consider $\tilde{R}_i(T) = (\gamma_i T^{e_i}, \tilde{\Gamma}_i(T) = \sum_{k=0}^{\tau_i} \tilde{\alpha}_{i,k} T^k)$ and denote $s_i = -v_T(\tilde{\Gamma}_i(T)) < 0$; we compute the inverse of $T^{s_i} \tilde{\Gamma}_i(T)$ (that has a non zero constant coefficient) via quadratic Newton iteration [18, Algorithm 9.3, page 259]; it takes less than $\mathcal{O}(\mathsf{M}(\tau_i + s_i))$ arithmetic operations [18, Theorem 9.4, page 260]. In order to get the singular part of the corresponding RPE R_i of F_∞, we need to know R_i with precision $\frac{r_i}{e_i}$, i.e. to know at least $r_i - s_i + 1$ terms. It is thus sufficient to know $\tilde{R}_i$ with precision $r_i - 2\, s_i$. This holds thanks to Proposition 12. Correctness and complexity of Algorithm `RNP` then follow straightforwardly from Propositions 9 and 10. □

Remark 8. Note that precision $v_X(\mathrm{Disc}_Y F)$ is not always enough to get the singular part of the Puiseux series centered at $(0,\infty)$, as shows the following example. Consider $F_3(X,Y) = 1 + X\, Y^{d-1} + X^{d+1}\, Y^d$. The singular parts of its RPEs are $(T, \frac{-1}{T^d})$ and $(-T^{d-1}, \frac{1}{T})$. Its reciprocal polynomial is $\tilde{F}_3 = Y^d + X\, Y + X^{d+1}$, with RPE's singular parts $(T,0)$ and $(-T^{d-1}, T)$. Here we have $v_X(\mathrm{Disc}_Y F) = d$, and $\lceil \tilde{F}_3 \rceil^d = Y^d + X\, Y$. The singular parts of $\lceil \tilde{F}_3 \rceil^d$ are indeed the same than the one of $\tilde{F}_3$, but we cannot recover the RPE $(T, \frac{-1}{T^d})$ of F from the RPE $(T,0)$ of $\lceil \tilde{F}_3 \rceil^d$. Nevertheless, the precision $\delta_{F_3} = v_X(\mathrm{lc}_Y(F_3)) + v_X(\mathrm{Disc}_Y F_3) = 2d+1$ is sufficient.

Proof of Theorem 5. Compute δ in less than $\mathcal{O}(\mathsf{M}(d_Y \delta) \log(d_Y \delta))$ operations from [24, Lemma 12], then run $\mathtt{RNP}(F, \delta)$ and conclude from Proposition 11. □

Remark 9. Another way to approach the non monic case is the one used in [31]. The idea is to use algorithms `MonicRNP` and `Half-RNP3` even when F is not monic. This would change nothing as far as these algorithms are concerned, but the proof concerning truncation bounds must be adapted:

1. define $s_i := \min(0, v_X(S_i))$, $N_i' := N_i - \frac{s_i}{e_i}\, d_Y$ and $v_i' := v_i - \frac{s_i}{e_i}\, d_Y$;
2. prove $N_i' = \frac{r_i}{e_i} + v_i'$ (use [31, Figure 3] for possible positive slopes of the initial call);
3. replace v_i by v_i' and N_i by N_i' in the remaining results of Section 3.3; proofs use some intermediate results of [31] (in particular, to prove $\frac{r_i}{e_i} \leq v_i'$, we need to use some formulæ in the proof of [31, Proposition 5, page 204]).

We chose to consider the monic case separately, since it makes one of the main technical results of this paper (namely tight truncation bounds) less difficult to apprehend, thus the paper more progressive to read.

5 Avoiding univariate factorisation.

We proved Theorem 1 up to the cost of univariate factorisations. To conclude the proof, one would additionally need to prove that the cost of all univariate factorisations computed when calling Algorithm `Half-RNP` is in $\tilde{\mathcal{O}}(\delta\, d_Y)$. As δ can be small, we would need a univariate factorisation algorithm for a polynomial in $\mathbb{K}[Y]$ of degree at most d with complexity $\tilde{\mathcal{O}}(d)$. Unfortunately, this does not exist. We will solve this point via Idea 4; relying on the "dynamic evaluation" technique [12, 13] (also named "D5 principle") of Della Dora, Dicrescenzo and Duval. This provides a way to compute with algebraic numbers, while avoiding factorisation (replacing it by *square-free* factorisation). In this context, we will consider polynomials with coefficients belonging to a direct product of field extensions of $\mathbb{K}$; more precisely to a zero-dimensional *non integral* $\mathbb{K}$-algebra $\mathbb{K}_I = \mathbb{K}[\underline{Z}]/I$, where I is defined as a triangular set in $\mathbb{K}[\underline{Z}] := \mathbb{K}[Z_1, \cdots, Z_s]$. As a consequence, zero divisors might appear, causing triangular decomposition and splittings (see Section 5.1 for details). Four main subroutines of the `Half-RNP` algorithm can lead to a decomposition of the coefficient ring:

(i) computation of Newton polygons,

(ii) square-free factorisations of characteristic polynomials,

(iii) subroutine `WPT`, via the initial gcd computation,

(iv) computation of primitive elements.

There are two other points that we need to take care of for our main program:

(v) subroutine `Hensel`, via the initial use of [24, Algorithm 1];

(vi) the initial factorisation of algorithm `RNP` (when computing Puiseux series above *all* critical points).

Remark 10. Dynamic evaluation is not the key point of this paper, and has been already considered for computing Puiseux series (see e.g. [13]). We could have simply said "split when required". However, keeping quasi-linear algorithms when dealing with dynamic evaluation in not an easy task, especially in our context where splittings may occur in many various subroutines. Hence, we decided to detail all steps and to be precise and self-contained about dynamic evaluation in our context. This makes this section relatively long and technical, but the reader may skip it at a first reading.

To simplify the comprehension of this section, we will not mention logarithmic factors in our complexity results, using only the $\mathcal{O}$ notation. This section is divided as follows:

1. We start by recalling a few definitions on triangular sets and in particular our notion of *D5 rational Puiseux expansions* in Section 5.1.
2. The key point of this section is to deal with these splitting with almost linear algorithms; to do so, we mainly rely on [12]. We briefly review in Section 5.2 their results; additionally, we introduce a few algorithms needed in our context. In particular, this section details points (iv) and (v) above.

3. Points (i) and (ii) above are grouped in a unique procedure `Polygon-Data`, detailed in Section 5.3.
4. We provide D5 versions of algorithms `Half-RNP`, `MonicRNP` and `RNP` in Section 5.4.
5. Finally, we prove Theorem 1 in Section 5.5.

5.1 Triangular sets and dynamic evaluation.

Definition 12. A (monic, autoreduced) *triangular set* of $\mathbb{K}[Z_1, \cdots, Z_s]$ is a set of polynomials $P_1, \cdots, P_s$ such that:

- $P_i \in \mathbb{K}[Z_1, \cdots, Z_i]$ is monic in Z_i,
- P_i is reduced modulo $(P_1, \cdots, P_{i-1})$,
- the ideal $(P_1, \cdots, P_s)$ of $\mathbb{K}[\underline{Z}]$ is radical.

We abusively call an ideal $I \subset \mathbb{K}[\underline{Z}]$ a triangular set if it can be generated by a triangular set $(P_1, \dots, P_s)$. We denote by $\mathbb{K}_I$ the quotient ring $\mathbb{K}[\underline{Z}]/(I)$.

Note that this defines a zero-dimensional lexicographic Gröbner basis for the order $Z_1 < \cdots < Z_s$ with a triangular structure. Such a product of fields contains zero divisor:

Definition 13. We say that a non-zero element $\alpha \in \mathbb{K}_I$ is *regular* if it is not a zero divisor. We say that a polynomial or a parametrisation defined over $\mathbb{K}_I$ is regular if all its non zero coefficients are regular.

Triangular decomposition. Given a zero divisor α of $\mathbb{K}_I$, one can divide I as $I = I_0 \cap I_1$ with $I_0 + I_1 = (1)$, $\alpha \mod I_0 = 0$ and $\alpha \mod I_1$ is invertible. Moreover, both ideals I_0 and I_1 can be represented by triangular sets of $\mathbb{K}[\underline{Z}]$.

Definition 14. A *triangular decomposition* of an ideal I is $I = I_1 \cap \cdots \cap I_k$ such that every I_i can be represented by a triangular set and $I_i + I_j = (1)$ for $1 \leq i \neq j \leq k$.

Thanks to the Chinese remainder theorem, the $\mathbb{K}$-algebra $\mathbb{K}_I$ is isomorphic to $\mathbb{K}_{I_1} \oplus \cdots \oplus \mathbb{K}_{I_k}$ for any triangular decomposition of I. We extend this isomorphism coefficient wise for any polynomial or series defined above $\mathbb{K}_I$.

Definition 15. Consider any polynomial or series defined above $\mathbb{K}_I$. We define its *splitting* according to a triangular decomposition $I = I_1 \cap \cdots \cap I_k$ the application of the above isomorphism coefficient-wise.

A key point (as far complexity is concerned) is the concept of *non critical* triangular decompositions. We recall [12, Definitions 1.5 and 1.6]:

Definition 16. Two polynomials $a, b \in \mathbb{K}_I[X]$ are said *coprime* if the ideal $(a, b) \subset \mathbb{K}_I[X]$ is equal to (1).

Definition 17. Let $(P_1, \cdots, P_s)$ and $(\tilde{P}_1, \cdots, \tilde{P}_s)$ be two distinct triangular sets. We define the *level* l of this two triangular sets to be the least integer such that $P_l \neq \tilde{P}_l$. We say that these triangular sets are *critical* if P_l and $\tilde{P}_l$ are not coprime in $\mathbb{K}[Z_1, \cdots, Z_{l-1}]/(P_1, \cdots, P_{l-1})$. A triangular decomposition $I = I_1 \cap \cdots \cap I_k$ is said *non critical* if it has no critical pairs ; otherwise, it is said *critical.*

D5 rational Puiseux expansions. We conclude this section by defining systems of D5-RPEs over fields and product of fields. Roughly speaking, a system of D5-RPE over a perfect field $\mathbb{K}$ is a system of RPEs over $\mathbb{K}$ grouped together with respect to some square-free factorisation of the characteristic polynomials, hence without being necessarily conjugated over $\mathbb{K}$. We have to take care of two main points:

1. We want correct informations (e.g. regularity indices) before fields splittings. To do so, the parametrisations we compute are regular (without any zero divisors).
2. We want to recover usual system of RPEs after fields splittings.

In particular, the computed parametrisations will fit the following definition:

Definition 18. Let $F \in \mathbb{K}[X, Y]$ be separable with $\mathbb{K}$ a perfect field. A system of *D5 rational Puiseux expansions* over $\mathbb{K}$ of F above 0 is a set $\{R_i\}_i$ such that:

- $R_i \in \mathbb{K}_{P_i}((T))^2$ for some square-free polynomial P_i,
- Denoting $P_i = \prod_j P_{ij}$ the univariate factorisation of P_i over $\mathbb{K}$ and $\{R_{ij}\}_j$ the splitting of R_i according to the decomposition $\mathbb{K}_{P_i} = \oplus_j \mathbb{K}_{P_{ij}}$, then the set $\{R_{ij}\}_{i,j}$ is a system of $\mathbb{K}$-RPE of F above 0 (as in Definition 2).

In order to deal with *all* critical points in Section 6, we will compute the RPE's of F above a root of a square-free factor Q of the resultant R_F:

Definition 19. Let $F \in \mathbb{K}_Q[X, Y]$ separable for some $Q \in \mathbb{K}[X]$ square-free. We say that F admits a system of D5-RPE's over $\mathbb{K}_Q$ above 0 if there exists parametrisations as in Definition 18 that are regular over $\mathbb{K}_Q$. Then, a system of *D5 rational Puiseux expansions* over $\mathbb{K}$ of F above the roots of Q is a set $\{Q_i, \mathcal{R}_i\}_i$ such that:

- $Q = \prod_i Q_i$,
- $\mathcal{R}_i$ is a system of D5 RPE's over $\mathbb{K}_{Q_i}$ of $F(X + z_i, Y)$ above 0 (in the sense of definition above), where z_i is the residue class of Z modulo $Q_i(Z)$.

5.2 Complexity of dynamic evaluation.

Results of [12]. We start by recalling the main results of [12], providing them only with the $\tilde{\mathcal{O}}$ notation (i.e. forgetting logarithmic factors). In particular, we will take $\mathsf{M}(d) \in \tilde{\mathcal{O}}(d)$ in the following. In our paper, we also assume the number of variables defining triangular sets to be constant (we usually have $s = 2$ in our context).

Definition 20. An *arithmetic time* is a function $I \mapsto \mathsf{A}_s(I)$ with real positive values and defined over all triangular sets in $\mathbb{K}[Z_1, \cdots, Z_s]$ such that:

1. For every triangular decomposition $I = I_1 \cap \cdots \cap I_h$, $\mathsf{A}_s(I_1) + \cdots + \mathsf{A}_s(I_h) \leq \mathsf{A}_s(I)$.
2. Any addition or multiplication in $\mathbb{K}_I$ can be made in $\mathsf{A}_s(I)$ operations over $\mathbb{K}$.
3. Given a triangular decomposition $I = I_1 \cap \cdots \cap I_h$, one can compute a non-critical triangular decomposition of I that refines it in less than $\mathsf{A}_s(I)$ arithmetic operations. We denote `removeCriticalPairs` such an algorithm.
4. Given $\alpha \in \mathbb{K}_I$ and a non-critical triangular decomposition $I = I_1 \cap \cdots \cap I_h$, one can compute the splitting of α in less than $\mathsf{A}_s(I)$ operations in $\mathbb{K}$. We denote `Split` such an algorithm.

Theorem 6. *Let $I = (P_1, \cdots, P_s)$ be a triangular set, and denote $d_i = \deg_{Z_i}(P_i)$. Assuming s to be constant, one can take $\mathsf{A}_s(I) \in \tilde{\mathcal{O}}(d_1 \cdots d_s)$*

Proof. This is a special case of the main result of [12], namely Theorem 8.1 therein. □

Proposition 13. *Let $I = (P_1, \cdots, P_s)$, and A, $B \in \mathbb{K}_I[Y]$ with degrees in Y less than d. Assuming s constant, one can compute the extended greatest common divisor of A and B in less than $\tilde{\mathcal{O}}(d \cdot d_1 \cdots d_s)$ operations over $\mathbb{K}$.*

Proof. This is [12, Proposition 4.1]. □

Splitting all coefficients of a polynomial. In the remaining of this section, we focus on the case $s = 2$, denoting $I = (Q, P)$, $d_Q = \deg_{Z_1}(Q)$, $d_P = \deg_{Z_2}(P)$ and $d_I = d_Q\, d_P$.

Lemma 12. *There exists an algorithm* `ReducePol` *that, given $H \in \mathbb{K}_I[X, Y]$, returns a collection $\{(I_k, H_k)_k\}$ such that $I = \cap_k I_k$ is a non critical triangular decomposition and the polynomials $H_k = H \mod I_k$ are regular over I_k. This algorithm performs at most $\tilde{\mathcal{O}}(\deg_X(H)\, \deg_Y(H)\, d_I)$ operations over $\mathbb{K}$.*

Proof. As for [12, Algorithm monic], for each coefficient of H, we split it according to the decomposition of I found so far. For each reduced coefficient we get, we test its regularity using gcd computation. This gives us a new (possibly critical) decomposition of I. We run Algorithm `removeCriticalPairs` on it. At the end, we split H according to the found decomposition. Complexity follows from Theorem 6 and Proposition 13. □

Square-free decomposition above $\mathbb{K}_I$. We say that a monic polynomial $\phi \in \mathbb{K}_I[Y]$ is square-free if the ideal $I + (\phi)$ is radical. $\phi = \prod_i \phi_i^{n_i}$ is the square-free factorisation of ϕ over $\mathbb{K}_I$ if the ϕ_i are coprime square-free polynomials in $\mathbb{K}_I[Y]$ and $n_i < n_{i+1}$ for all i.

Proposition 14. *Consider $\mathbb{K}$ a perfect field with characteristic p and $\phi \in \mathbb{K}_I[Y]$ a monic polynomial of degree d. Assuming $p = 0$ or $p > d$, there exists an algorithm* `SQR-Free` *that computes a set* $\{(I_k, (\phi_{k,l}, M_{k,l})_l)_k\}$ *such that* $I = \cap_k I_k$ *is a non critical triangular decomposition and* $\phi_k = \prod_l \phi_{k,l}^{M_{k,l}}$ *is the square-free factorisation of* $\phi_k := \phi \mod I_k$*. It takes less than* $\tilde{\mathcal{O}}(d\, d_I)$ *operations over* $\mathbb{K}$*.*

Proof. We compute successive gcds and euclidean divisions, using Yun's algorithm [18, Algorithm 14.21, page 395] (this result is in characteristic 0, but works in positive characteristic when $p > d$). Each gcd computation is Proposition 13. We just need to add splitting steps (if needed) in between two calls. The complexity follows by using Proposition 13 in the proof of [18, Theorem 14.23, page 396], since there are less than d calls to the algorithm `removeCriticalPairs`. $\square$

Keeping a constant number of variables. We extend the result of Proposition 4 above $\mathbb{K}_Q$ for some square-free polynomial Q. This requires additional attention on splittings.

Proposition 15. *Let* $\phi \in \mathbb{K}_I[Z_3]$ *square-free,* $d = d_P \deg_{Z_3}(\phi)$*. If* $\mathbb{K}$ *contains at least* d^2 *elements, there exists a Las-Vegas algorithm that computes* $(Q_k, P'_k, \psi_k)_k$ *satisfying:*

- $Q = \prod_k Q_k$*,*
- P'_k *is a squarefree polynomial of degree* d *over* $\mathbb{K}_{Q_k}$*,*
- $\psi_k : \mathbb{K}_{I_k} \to \mathbb{K}_{I'_k}$ *is an isomorphism, where* $I_k = (Q_k, P, \phi)$ *and* $I'_k = (Q_k, P'_k)$*.*

We call `BivTrigSet` *such an algorithm. It takes* $\tilde{\mathcal{O}}(d^{\frac{\omega+1}{2}}\, d_Q)$ *operations over* $\mathbb{K}$*. Given* $H \in \mathbb{K}_{I_k}[X, Y]$*, one can compute* $\psi_k(H)$ *in less than* $\tilde{\mathcal{O}}(\deg_X(H)\, \deg_Y(H)\, d_P\, d\, d_{Q_k})$*.*

Proof. We follow the Las Vegas algorithm[3] given in [34, Section 2.2]. First, trace computation of the monomial basis takes $\mathcal{O}(\mathsf{M}(d\, d_Q))$ operations in $\mathbb{K}$ (it is reduced to polynomial multiplication thanks to [27, Proposition 8]). Then, picking a random element A, we compute the $2\,d$ traces of powers of A by power projection. Methods based on [36] involve only polynomial, transposed polynomial and matrix multiplications, for a total in $\mathcal{O}(d^{\frac{\omega+1}{2}}\mathsf{M}(d_Q))$ operations in $\mathbb{K}$. Finally, our candidate for P' can be deduced via Newton's method in $\mathcal{O}(\mathsf{M}(d\, d_Q))$ operations. It remains to test its square-freeness, involving gcd over $\mathbb{K}_Q$. It takes less than $\tilde{\mathcal{O}}(d\, d_Q)$ operations over $\mathbb{K}$ from Proposition 13. If a factorisation of Q appears, we run some splittings and Theorem 6 concludes.

To compute ψ_k, we first need d additional traces; this is once again power projection. Then, one solves a linear system defined by a Hankel matrix (see [36, Proof of Theorem 5]). This can be done using the algorithm described in [7], that reduces the problem to extended gcd computation, thus involves potential decomposition of Q. This is once again $\tilde{\mathcal{O}}(d\, d_Q)$ operations over $\mathbb{K}$ (using `removeCriticalPairs` if needed).

To conclude, using e.g. Horner's scheme [31, Section 5.1.3, page 209], rewriting the coefficients of $H \in \mathbb{K}_{I_k}[X, Y]$ can be done in $\tilde{\mathcal{O}}(\deg_X(H)\, \deg_Y(H)\, d_P\, d\, d_{Q_k})$. $\square$

[3] here the assumption on the number of elements of $\mathbb{K}$ is used

Remark 11. Algorithm `BivTrigSet` keeps the number of variables constant (at most two) for the triangular sets we are using during the whole algorithm. We do not work with univariate triangular sets for two reasons:

1. Computing such triangular set (starting from a bivariate one) would lead to a bound in $d_Q^{\frac{\omega+1}{2}}$, that can be $D^{\omega+1}$ when the factor Q of the resultant has high degree (see Section 6). As $\omega > 2$, this is too much.

2. Q (factor of the resultant) and P (residual extension) do not provide the same geometrical information.

Extending `WPT` and `Hensel` to the D5 context. We conclude this section by providing trivial extension of the Hensel algorithms: we only need to pay attention to the initial gcd-computation (for `WPT`) or its generalised version of [24] (for `Hensel`).

Proposition 16. *Let $G \in \mathbb{K}_I[X,Y]$ and $n \in \mathbb{N}$. There exist an algorithm that computes a set $(I_k, \lceil \widehat{G}_k \rceil^n)$ such that $I = \cap_k I_k$ is a non critical decomposition of I and $\widehat{G}_k$ the Weierstrass polynomial of $G \mod I_k$. It takes less than $\mathcal{O}(\mathsf{M}(n \deg_Y(G)\, d_I))$ operations in $\mathbb{K}$. We still denote `WPT` such an algorithm.*

Proof. First run `ReducePol` if needed (it is not in our context), getting a set (I_i, G_i'). Than, for each i, use extended Euclidean algorithm with parameters $(Y^{M_i}, Y^{-M_i}\, G_i(0,Y))$ with $M_i = \upsilon_Y(G_i(0,Y)$, getting a decomposition $I_i = \cap_j I_{ij}$ and associated Bézout relations. Compute a non triangular decomposition $I = \cap_k I_k$ that refines $\cap_i \cap_j I_{ij}$, and reduce G and the Bézout relations accordingly. Finally, run the Hensel lemma (that does not generate any splitting) on each G_k, using the associated Bézout relation. Complexity follows from Lemma 12, Proposition 13, Theorem 6 and Proposition 3. ☐

Lemma 13. *Given G, $H \in \mathbb{K}_I[X,Y]$ of degrees in Y bounded by d, one can compute a set $(I_k, G_k, H_k, U_k, V_k, \eta_k)_k$ such that $I = \cap_k I_k$ is a non critical decomposition of I, $G_k = G \mod I_k$, $H_k = H \mod I_k$ and $U_k \cdot G_k + V_k \cdot H_k = X^{\eta_k} \mod X^{\eta_k+1}$ with η_k the lifting order of (G_k, H_k). This takes $\tilde{\mathcal{O}}(d\, d_I \max_k \eta_k)$ operations over $\mathbb{K}$.*

Proof. As said in the introduction of their paper, [24, Algorithm 1] is "a suitable adaptation of the half-gcd algorithm": a call to their algorithm uses polynomial multiplication (more precisely multiplications of 2×2 matrices of univariate polynomials), two recursive calls and one computation of the "pseudo-division operator" $\mathcal{Q}$ [24, Section 3.1], which includes euclidean division, extended Euclidean algorithm and Hensel lifting ([25, Algorithm Q] to compute "normal form" of polynomials). Whence a finite number of call that induce splittings, all considered in [12] (multiplication induces no splitting, Euclidean algorithm is the key point of [12], and [25, Algorithm Q] induces splitting only once, via the extended Euclidean algorithm). ☐

Proposition 17. *Let $n \in \mathbb{N}$, F, G, $H \in \mathbb{K}_I[X,Y]$ with H monic in Y, $F = G\,H \mod X^{2\,\eta+1}$ and $\eta \geq \kappa(G,H)$. There exists an algorithm that computes a set $\{I_k, G_k, H_k\}_k$*

such that $I = \cap_k I_k$ is a non critical decomposition of I, $G_k = G \mod (I_k, X^{\eta_k+1})$, $H_k = H \mod (I_k, X^{\eta_k+1})$ and $F \mod I_k = G_k H_k \mod X^{n+2\eta_k}$, where $\eta_k = \kappa(G_k, H_k)$. Moreover, if $G_k^\star$, $H_k^\star \in \mathbb{K}_{I_k}[X,Y]$ satisfy $F \mod I_k = G_k^\star H_k^\star \mod X^{n+2\eta_k}$, then $G_k = G_k^\star \mod X^n$ and $H_k = H_k^\star \mod X^n$. It takes less than $\mathcal{O}(\mathsf{M}(n\, d_Y\, d_I))$ operations in $\mathbb{K}$. We still denote `Hensel` *such an algorithm.*

Proof. The D5 adaptation of the `Hensel` algorithm is straightforward: use Lemma 13 first, then run `HenselStep` as many times as necessary for each $(I_i, G_i, H_i, U_i, V_i, \kappa_i)$ you get, as in the proof of Lemma 10. $\square$

5.3 Computing polygon datas in the D5 context.

To simplify the writing of the `Half-RNP3` algorithm, we group in algorithm `Polygon-Data` below the computation of the Newton polygon and the square-free decomposition of associated characteristic polynomials. Given $H \in \mathbb{K}_I[X,Y]$ known with precision n, it returns a list $\{(I_i, H_i, \Delta_{ij}, \phi_{ijk})\}_k$ such that:

- $I = \cap I_i$ is a non critical triangular decomposition;
- $H_i := H \mod I_i$ is regular;
- $\mathcal{N}_n(H_i) = \{\Delta_{ij}\}_j$;
- $\prod_k \phi_{ijk}^{M_{ijk}}$ is the square-free factorisation of $\phi_{\Delta_{ij}}$.

Algorithm: `Polygon-Data`(H, I, n)

In: I a bivariate triangular set and $H \in \mathbb{K}_I[X,Y]$ known modulo X^{n+1}. We assume $n > 0$ and $\deg_Y(H) > 0$.

Out: A list $\{(I_i, H_i, \Delta_{ij}, \phi_{ijk}, M_{ijk})\}$ as explained above.

foreach (H_i, I_i) **in** `ReducePol`(H, I) **do**
  $\{\Delta_{ij}\}_{j=1,\dots,s} \leftarrow \mathcal{N}_n(H_i)$; // H_i is regular
  for $j = 1, \dots, s$ **do**
    $\{I_i^l, \phi_{ijk}^l, M_{ijk}^l\} \leftarrow$ `SQR-Free`$(\phi_{\Delta_{ij}}, I_i)$
$\{I'_h\}_h \leftarrow$ `removeCriticalPairs`$(\{I_i^l\}_{i,l})$;
$\{H'_h\}_h \leftarrow$ `Split`$(H_i, \{I_i^l\}_{i,l}, \{I'_h\}_h)$;
foreach i, j, k **do**
  $\{\phi'_{mjk}\}_{mjk} \leftarrow$ `Split`$(\phi_{ijk}^l, \{I_i^l\}_l, \{I'_h\}_h)$; // taking the right subset $\{I'_h\}_h$
return $\{(I'_h, H'_h, \Delta_{i(m)j}, \phi'_{mjk})\}_{m,j,k}$; // $i(m) : m \mapsto$ correct i

Proposition 18. *Algorithm* `Polygon-Data` *is correct and takes $\tilde{\mathcal{O}}(\deg_X(H)\, \deg_Y(H)\, d_I)$ operations in $\mathbb{K}$.*

Proof. Exacteness and complexity follow from Proposition 14 and Theorem 6, using $\sum_{j,k,l} \deg(\phi_{ijk}^l) \le d_Y(H)$ for all i and $\sum_{i,l} \deg(I_i^l) = \sum_h \deg(I'_h) = \sum_i \deg(I_i) = d_I$. $\square$

5.4 Computing half Puiseux series using dynamic evaluation.

In order to compute also the RPEs of F above the roots of any squarefree factor Q of the resultant, we are led to consider $I = (Q, P)$ instead of P as an input for `Half-RNP3`, the D5 variant of `Half-RNP`. More precisely, the input is a set H, I, n, π such that:

- $I = (Q, P)$ is a bivariate triangular set over $\mathbb{K}$ ($P = Z_2$ initially, $Q = Z_1$ admitted);
- $H \in \mathbb{K}_I[X, Y]$ separable, monic in Y, with $d := \deg_Y(H) > 0$;
- $n \in \mathbb{N}$ is the truncation order we will use for the powers of X during the algorithm;
- π the current truncated parametrisation ($\pi = (X, Y)$ for the initial call).

The output is a set $\{I_i, \mathcal{R}_i\}_i$ such that:

- $I = \cap_i I_i$ is a non critial decomposition,
- $\mathcal{R}_i = \{R_{ij}\}$ is a set of D5-RPE's of $H_i := H \mod I_i$ satisfying $n - v_{ij} \geq r_{ij}$ and given with precision at least $(n - v_{ij})/e_{ij} \geq r_{ij}/e_{ij} \geq 0$,

where we let $v_{ij} := v_X\left(\partial_Y H_i(S)\right)$ for any Puiseux series S associated to R_{ij}. We refer to the field version `Half-RNP` for all notations which are not specified here.

Algorithm: `Half-RNP3`(H, I, n, π)

$B \leftarrow A_{d-1}/d$; $\pi' \leftarrow \lceil \pi(X, Y - B) \rceil^n$; // $H = \sum_{i=0}^{d} A_i Y^i$
if $d = 1$ **then return** $(I, \pi'(T, 0))$ **else** $H' \leftarrow \lceil H(X, Y - B) \rceil^n$;
$(I_i, H_i, \Delta_i, \phi_i)_i \leftarrow$ `Polygon-Data`(H', I, n);
$\{\pi_i\}_i \leftarrow$ `Split`$(\pi', \{I_i\}_i)$; // taking only once each different I_i
forall i **do**
    if $\deg(\phi_i) = 1$ **then** $\xi_{i1}, I_{i1}, H_{i1}, \pi_{i1} = -\phi_i(0), I_i, H_i, \pi_i$;
    else
        $\{I_{ij}, \Psi_{ij}\}_j \leftarrow$`BivTrigSet`$(I_i, \phi_i)$;
        $\{H'_{ij}\}_j \leftarrow$ `Split`$(H_i, \{I_{ij}\}_j)$; $\{\pi'_{ij}\}_j \leftarrow$ `Split`$(\pi_i, \{I_{ij}\}_j)$;
        forall j **do** $\xi_{ij}, H_{ij}, \pi_{ij} \leftarrow \Psi_{ij}(Z), \Psi_{ij}(H'_{ij}), \Psi_{ij}(\pi'_{ij})$;
    forall j **do** // Δ_i belongs to $m_i\, a + q_i\, b = l_i$; $u_i, v_i =$ `Bézout`(m_i, q_i)
        $\pi''_{ij} \leftarrow \pi_{ij}(\xi_{ij}^{v_i}\, X^{q_i}, X^{m_i}\,(Y + \xi_{ij}^{u_i})) \mod I_{ij}$;
        $H''_{ij} \leftarrow \lceil H_{ij}(\xi_{ij}^{v_i}\, X^{q_i}, X^{m_i}\,(Y + \xi_{ij}^{u_i})) \rceil^{n_i} \mod I_{ij}$; // $n_i = q_i\, n - l_i$
        $\{(I_{ijk}, H_{ijk})\} \leftarrow$ `WPT`(H''_{ij}, n_i);
        $\pi_{ijk} \leftarrow$ `Split`$(\pi''_{ij}, \{I_{ijk}\}_{ijk})$;
        forall k **do** $\{I_{ijkl}, \mathcal{R}_{ijkl}\}_l \leftarrow$ `Half-RNP3`$(H_{ijk}, I_{ijk}, n_i, \pi_{ijk})$;
$\mathcal{R} \leftarrow \{\}$; $\{I'_h\}_h \leftarrow$ `removeCriticalPairs`$(\{I_{ijkl}\}_{ijkl})$;
forall i, j, k, l **do** // taking the subset of $\{I'_h\}_h$ refining I_{ijkl}
    $\mathcal{R} \leftarrow \mathcal{R} \cup$ `Split`$(\mathcal{R}_{ijkl}, \{I'_h\}_h)$
return $\mathcal{R}$; // each element of $\mathcal{R}$ coupled to their associated I'_h

Proposition 19. *Let $Q \in \mathbb{K}[Z]$ be square-free and $F \in \mathbb{K}_Q[X,Y]$ be monic and separable in Y. The function call* Half-RNP3$(F,(Q,Z),n,(X,Y))$ *returns a correct answer in an expected $\tilde{\mathcal{O}}(d_Q\, n\, d_Y{}^2)$ operations over $\mathbb{K}$.*

Proof. Just adapt the proof of Proposition 7 to the D5 context, using Propositions 15, 16 and 18, together with Theorem 6. □

5.5 Proof of Theorem 1.

We finally conclude the proof of Theorem 1, providing the D5 variants of algorithms MonicRNP and RNP, namely algorithms MonicRNP3 and RNP3 below.

The monic case. As in Section 4, we begin with the monic case. Therein, we assume that the Hensel algorithm is a D5 version, as explained in Section 5.2. Also, we recall that v_{ij} denotes $v_X(\partial_Y H_i(S))$ for any Puiseux series S associated to R_{ij}.

Algorithm: MonicRNP3(F,Q,n)

In: $Q \in \mathbb{K}[Z]$ square-free, $F \in \mathbb{K}_Q[X,Y]$ separable and monic in Y, and $n \in \mathbb{N}$.
Out: $\{(Q_i,\mathcal{R}_i)\}_i$, with $Q = \prod Q_i$ and $\mathcal{R}_i$ a system of singular parts of D5-RPEs of $F \mod Q_i$ above 0.

$\eta \leftarrow \min(n, 6\,n/d_Y)$; $\mathcal{R} \leftarrow \{\}$;
$\{I_i,\mathcal{R}_i\}_i \leftarrow$ Half-RNP3$(F,(Q,Z_2),\eta,\pi)$; // $I_i = (Q_i, Z_2)$
$\{F_i\}_i \leftarrow$ Split$(F,\{Q_i\}_i)$;
forall i **do**
  Keep in $\mathcal{R}_i$ the R_{ij} such that $v_{ij} < \eta/3$; // known with precision $\geq 2\eta/3$
  if $\#\mathcal{R}_i = d_Y$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \{Q_i,\mathcal{R}_i\}$; **continue**;
  $G_i \leftarrow$ NormRPE$(\mathcal{R}_i, 2\eta/3)$;
  $H_i \leftarrow$ Quo$(F_i, G_i, 2\eta/3)$; // no splitting since G_i is monic
  $\{Q_{ij}, G_{ij}, H_{ij}\}_j \leftarrow$ Hensel(F_i,G_i,H_i,n);
  forall j **do** $\{(Q_{ijk},\mathcal{R}_{ijk})\}_k \leftarrow$ MonicRNP3(H_{ij},Q_{ij},n,π);
  $\{\mathcal{R}'_{ijk}\} \leftarrow$ Split$(\mathcal{R}_i,\{Q_{ijk}\}_{j,k})$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(Q_{ijk},\mathcal{R}_{ijk} \cup \mathcal{R}'_{ijk})_{j,k}\}$;
return $\mathcal{R}$

Recall the notations $R_F = \mathrm{Res}_Y(F, F_Y)$ and $\delta = v_X(R_F)$. We obtain:

Proposition 20. *Assuming that $n \geq \delta$ and that the trailing coefficient of R_F is not a zero divisor in $\mathbb{K}_Q$, a function call* MonicRNP3(F,Q,n) *returns a correct answer in an expected $\tilde{\mathcal{O}}(d_Q\, d_Y\, n)$ operations over $\mathbb{K}$.*

Proof. The assumption on the trailing coefficient of the resultant of F is needed only to ensure that the truncation bound δ is enough over all factors of Q. Otherwise, this is just an adaptation of the proof of Proposition 9 to the D5 context, using Propositions 17 and 19, together with Theorem 6 once again (subroutine `Quo` is used only with monic polynomials, and the remaining operations do not include any division). □

The general case. Algorithm `RNP3` below computes a system of singular part (at least) of D5-RPEs of a primitive polynomial F above the roots of any square-free factor Q of its resultant R_F. We follow the same strategy as in Algorithm `RNP`, but we take care of triangular decompositions due to division by zero divisors. In particular, we assume that algorithm `Monic` is a D5 version (it contains one call to the extended Euclidean algorithm). Also, inversion of the RPEs of $\tilde{F}_\infty$ can lead to some splittings (while inverting the trailing coefficient of the series). However, we do not detail these further splittings for readibility.

Algorithm: `RNP3`(F, Q, n)

In: $Q \in \mathbb{K}[Z_1]$ square-free, $F \in \mathbb{K}[X, Y]$ separable in Y with $d_Y > 0$, and $n \in \mathbb{N}$ big enough.

Out: A system of singular parts (at least) of D5-RPEs of F above the roots of Q.

$\mathcal{R} \leftarrow \{\}$; $\tilde{F} \leftarrow \lceil F(X + Z_1, Y) \mod Q\rceil^n$; // thus $\tilde{F} \in \mathbb{K}_Q[X, Y]$
$\{Q_i, F_{i,0}, F_{i,\infty}\}_i \leftarrow$ `Monic`$(\tilde{F}, n)$;
forall i **do**
  $\tilde{F}_{i,\infty} \leftarrow Y^{\deg_Y(F_{i,\infty})} F_{i,\infty}(X, 1/Y)$;
  $\{Q_{ij}, \mathcal{R}_{ij}\}_j \leftarrow$ `MonicRNP3`$(F_{i,0}, Q_i, n)$;
  $\{Q'_{ik}, \mathcal{R}'_{ik}\}_k \leftarrow$ `MonicRNP3`$(\tilde{F}_{i,\infty}, Q_i, n)$;
  forall k **do**
    Inverse the second element of each $R \in \mathcal{R}'_{ik}$;
    Split $\{Q'_{ik}, \mathcal{R}'_{ik}\}$ if required;
  $\{Q''_{il}\}_l \leftarrow$ `removeCriticalPairs`$(\{Q_{ij}\}_j \cup \{Q'_{ik}\}_k)$;
  forall k, j **do** $\mathcal{R} \leftarrow \mathcal{R} \cup$ `Split`$(\mathcal{R}_{ij},\{Q''_{il}\}_l) \cup$ `Split`$(\mathcal{R}'_{ik},\{Q''_{il}\}_l)$;
return $\mathcal{R}$; // elements of $\mathcal{R}$ with the same Q''_{il} grouped together

Proposition 21. *Assuming that Q is a square-free factor of R_F with multiplicity $n_Q \leq n$, a function call `RNP3`(F, Q, n) returns the correct answer in less than $\tilde{\mathcal{O}}(d_Q\, d_Y\, n)$ operation overs $\mathbb{K}$.*

Proof. The correctness follows from Propositions 11 and 20 (the trailing coefficient of the resultant of $F_{i,0}$ and $F_{i,\infty}$ is not a zero divisor by construction). The complexity follows from Propositions 2 and 20, Theorem 6, together with the relations $\deg_Y(F_{i,0}) + \deg_Y(F_{i,\infty}) = d_Y$ and $\sum_i \deg(Q_i) = d_Q$. □

Proof of Theorem 1. The algorithm mentionned in Theorem 1 is Algorithm `RNP3`, run with parameters $Q = Z_1$ and $n = \delta$, which can be computed via [24, Algorithm 1] in the aimed bound. Note that as we consider the special case $Q = Z_1$, F has coefficients over a field and this operation does not involve any dynamic evaluation. The function call $\texttt{RNP3}(F, Z_1, \delta)$ fits into the aimed complexity thanks to Proposition 21. □

6 Desingularisation and genus of plane curves.

It is now straightforward to compute a system of singular parts of D5 rational Puiseux expansions above all critical points. We include the RPEs of F above $x_0 = \infty$, defined as RPEs above $x_0 = 0$ of the reciprocal polynomial $\tilde{F} := X^{d_X} F(X^{-1}, Y)$. We have $\upsilon_X(R_{\tilde{F}}) = d_X\,(2\,d_Y - 1) - \deg(R_F)$.

Definition 21. Let $F \in \mathbb{K}[X, Y]$ be a separable polynomial over a field $\mathbb{K}$. A D5-desingularisation of F over $\mathbb{K}$ is a collection $\{(\mathcal{R}_1, Q_1), \ldots, (\mathcal{R}_s, Q_s), \mathcal{R}_\infty\}$ such that:

- $Q_k \in \mathbb{K}[X]$ are pairwise coprime, square-free and satisfy $R_F = \prod_{k=1}^{s} Q_k^{n_k}$, $n_k \in \mathbb{N}^*$;
- $\mathcal{R}_k$ is a system of singular parts (at least) of D5-RPEs of F above the roots of Q_k;
- $\mathcal{R}_\infty$ is a system of singular parts (at least) of D5-RPEs of F above $X = \infty$.

Note the following points:

- we can deduce from a D5-desingularisation of F the singular part of the RPE's of F above any root of R_F,
- we allow $n_k = n_l$ for $k \neq l$ (the factorisation $R_F = \prod_{k=1}^{s} Q_k^{n_k}$ is not necessarily a square-free factorisation).

We obtain the following algorithm:

Algorithm: $\texttt{Desingularise}(F)$
In: $F \in \mathbb{K}[X, Y]$ separable and primitive in Y, with $d_Y > 0$.
Out: The D5-desingularisation of F over $\mathbb{K}$

$\mathcal{R} \leftarrow \{\}$;
forall $(Q, n) \in \texttt{SQR-Free}(R_F)$ **do** $\mathcal{R} \leftarrow \mathcal{R} \cup \texttt{RNP3}(F, Q, n)$;
$n \leftarrow d_X\,(2\,d_Y - 1) - \deg(R_F)$;
if $n > 0$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \texttt{RNP3}(\lceil X^{d_X} F(X^{-1}, Y) \rceil^n, Z, n)$;
return $\mathcal{R}$

Proposition 22. *Algorithm* $\texttt{Desingularise}(F)$ *works as specified. It takes an expected* $\mathcal{O}^{\sim}(d_X\, {d_Y}^2)$ *operations over* $\mathbb{K}$.

Proof. Correctness is straightforward from Proposition 21. The computation of the resultant R_F fits in the aimed bound [18, Corollary 11.21, page 332], so is its square-free factorisation [18, Theorem 14.20, page 4]. The complexity is then a consequence of Proposition 21, using the classical formula $\sum_k \deg(Q_k) n_k + \delta_{\tilde{F}} = d_X\,(2\,d_Y - 1)$. □

Proof of Theorem 2. It follows immediately from Proposition 22. $\square$

Computing the genus of plane curves: proof of Corollaries 1, 2 and 3. Let $\{(Q_k, \mathcal{R}_k)\}_k$ be a D5-desingularisation of F. Since the D5-RPEs $R_{ki} \in \mathcal{R}_k$ are regular by construction, the ramification indices of all classical Puiseux series (i.e with coefficients in $\overline{\mathbb{K}}$) determined by R_{ki} are equal. If F is irreducible over $\overline{\mathbb{K}}$, the Riemann-Hurwitz formula determines the genus g of the projective plane curve defined by F as

$$g = 1 - d_Y + \frac{1}{2}\sum_k \deg(Q_k) \sum_{i=1}^{\rho_k} f_{ki}(e_{ki} - 1),$$

where f_{ki} and e_{ki} are respectively the residual degrees and ramification indices of the RPE R_{ki}. This proves Corollary 1. Corollaries 2 and 3 follow from [30, 32], where the authors show that we can reduce F modulo a well chosen small prime within the given bit complexities.

7 Factorisation in $\mathbb{K}[[X]][Y]$.

Our aim is to compute the irreducible analytic factors of F in $\mathbb{K}[[X]][Y]$ with precision X^N, and to do so in at most $\tilde{\mathcal{O}}(d_Y(\delta + N))$ operations over $\mathbb{K}$, plus the cost of one univariate factorisation of degree at most d_Y. The idea is to first compute a factorisation modulo X^δ, and then to lift this factorisation thanks to the following result:

Proposition 23. *Let $F \in \mathbb{K}[[X]][Y]$, separable of degree d. Suppose given a modular factorisation*

$$F \equiv uF_1 \cdots F_k \mod X^n, \quad n > 2\kappa \tag{1}$$

where $u \in \mathbb{K}[[X]]^\times$, for all i either F_i or its reciprocal polynomial $\tilde{F}_i$ is monic, and

$$\kappa = \kappa(F_1, \ldots, F_k) := \max_{I,J} \kappa(F_I, F_J),$$

the maximum of the lifting orders being taken over all disjoint subsets I, $J \subset \{1, \ldots, k\}$, with $F_I = \prod_{i\in I} F_i$. Then there exists uniquely determined analytic factors $F_1^*, \ldots, F_k^*$ such that $F = u^* F_1^* \cdots F_k^*$, where

$$F_i^* \equiv F_i \mod X^{n-\kappa} \quad \text{and} \quad u^* \in \mathbb{K}[[X]],\ u^* \equiv u \mod X^{n-\kappa}.$$

Moreover, starting from (1), we can compute the F_i^* up to an any precision $N \geq n - \kappa$ in $\tilde{\mathcal{O}}(dN)$ operations over $\mathbb{K}$.

Proof. Replace in [18, Algorithm 15.17] the use of [18, Algorithm 15.10] (line 6) by the `HenselStep` algorithm, and the extented Euclidean algorithm (line 4) by [24, Algorithm 1]. Existence and unicity of the lifting follow from Lemma 10. So does complexity. $\square$

Remark 12. This results improves [10, Lemma 4.1], where κ is replaced by $\delta/2 \geq \kappa$. Note that if $\kappa = 0$, this is the classical multifactor Hensel lifting. Otherwise, note that instead of starting from a univariate factorisation, we need to know the initial factorisation modulo a higher power of X.

Proof of Theorem 3. We proceed as follows:

1. Compute δ in the aimed bound.
2. Adapt `RNP3` (called with parameters F, Z and δ):
   - Make the `NormRPE` call (line 7 of `MonicRNP3`) additionally output minimal polynomials of the computed RPEs (i.e. the polynomials G_i of Section 4.1);
   - Replace the `Hensel` call (line 9 of `MonicRNP3`) by its multi-factor version (i.e. Proposition 23);
   - Output the lifted factors instead of the RPEs in `MonicRNP3`.
3. We get factors $\tilde{F}_i$ known modulo $X^{\delta+1}$, with coefficients in a product of fields $\mathbb{K}_{P_i}$ and $\sum \deg(P_i) = \sum f_i \leq d_Y$. Perform the univariate factorisation of the P_i and split accordingly the $\tilde{F}_i$ to get a factorisation $F = u^* F_1^* \cdots F_k^*$ modulo X^δ.
4. If $n > \delta$, use Proposition 23 to lift this factorisation to the required precision. □

8 Concluding remarks

In this paper, we provide worst-case complexity bounds for the local and global desingularisation which are equivalent (up to a logarithmic factor) to the computation of respectively the first non-zero coefficient of the resultant R_F [24] and the resultant computation. However, this provides for the moment only a theoretical algorithm: our algorithm is a combination of many subroutines, and the implementation of a fast efficient version would require a huge amount of work, especially due to the dynamic evaluation part. Moreover, there might be algorithm easier to implement that we plan to study in future work (see below).

Worst case complexity is sharp. We begin this section by providing a family of polynomial for which our complexity bounds are reached.

Example 5. Let $d > 3$ be divisible by 2 and consider $F = Y^d + (Y - X^{d/2})^2$, so that $d_X = d_Y = D = d$. By Hensel's lemma, we have $F = G\,H \in \mathbb{Q}[[X]][Y]$ with $G(0, Y) = Y^{d-2}+1$ and $H(0, Y) = Y^2$. As $G(0, Y)$ is square-free, we deduce immediately the singular parts of the Puiseux series of G (that is, their constant term here). In order to compute the singular parts of (at least half) the Puiseux series of H above 0 using algorithm `RNP3`, we need to lift further the factorisation $F = G\,H \mod X$ up to precision $\sigma \in \Theta(\delta_H/\deg_Y(H))$, and this precision is sharp from Lemma 6. We have $\delta_H = \delta = d^2$

while $\deg_Y(H) = 2$ is constant. Hence the required precision is in $\Theta(d^2)$ and the lifting step costs $\Theta(d^3) = \Theta(D^3)$, leading to a cubic complexity in the total degree.

Irreducibility test via `Half-RNP` is $\Omega(d_Y\,\delta)$. The previous example shows the sharpness of the divide and conquer strategy. But even the first step (algorithm `Half-RNP3`) is sharp, due to the "blowing up" of the Puiseux transform. As a consequence, even for an irreducible polynomial (where there is no need of the divide and conquer strategy), complexity of Theorem 1 is sharp, as shows the following example:

Example 6. Let $d > 12$ be divisible by 4 and consider F to be the minimal polynomial of the Puiseux series $S(X) = X^{\frac{4}{d}} + X + X^{\frac{d+1}{d}}$. We have $d_Y = d$, $\delta = 7\,d - 13$ and $v_X(F_Y(S)) = 7 - \frac{13}{d}$, and Lemma 6 proves that we need to consider $\lceil F \rceil^n$ with $n = 8 - \frac{d}{12} > \frac{\delta}{d}$, i.e. $F \mod X^8$. We have $\mathcal{N}_n(F) = ((0,4),(d,0))$ with characteristic polynomial $(T-1)^4$, so that $m_1 = 1$, $q_1 = \frac{d}{4}$ and $l_1 = d$. We therefore need to compute the Puiseux transform $G(X,Y) = \lceil F(X^{\frac{d}{4}}, X\,(Y+1))/X^d \rceil^{n_1}$ with $n_1 = \frac{d}{4}\,n - d = d - 3$. As G has size $d\,n_1 \in \Omega(d_Y\,\delta)$, so is the complexity of Lemma 2, thus of Theorem 1.

As a consequence, this blowing-up step prevents any Newton–Puiseux like method for providing an irreducibility test in $\mathbb{K}[[X]][Y]$ (or $\overline{\mathbb{K}}[[X]][Y]$) in $\tilde{\mathcal{O}}(\delta)$ operations in $\mathbb{K}$. We plan to investigate the approach of Abhyankhar [2] to improve that point; in particular, we hope such an approach to improve the practical implementation of the algorithm.

The reverse role strategy. If we only want Puiseux series centered at $(0,0)$, we can try to invert the roles played by X and Y: thanks to the inversion formula [17, Proposition 4.2], we can recover the singular parts of the Puiseux series of F centered at $(0,0)$ with respect to Y from those of $\tilde{F}(X,Y) = F(Y,X)$.

Considering Example 5, the polynomial $\tilde{F} \in \mathbb{K}[[X]][Y]$ is then Weierstrass of degree d. One can compute $\delta_{\tilde{F}} = d^2 + 2\,(d-1)$. Hence, we need a lifting precision $\tilde{\sigma} \in \Theta(\delta_{\tilde{F}}/d) = \Theta(d)$ in order to compute at least half of the Puiseux series of $\tilde{F}$, for a total cost $\Theta(d^2)$. As $\tilde{F}$ has edge polynomial $(Y^{d/2} - X)^2$, we deduce that we will in fact separate the singular parts of *all* Puiseux series of $\tilde{F}$ with precision $\tilde{\sigma}$ - recovering then those of F by appyling the inversion formula - for a total quadratic cost $\Theta(d^2) = \Theta(D^2)$ assuming that we may apply the inversion formula within this bound.

Remark 13. We did not check that applying the inversion formula really fits in the aimed bound. This problem is closely related to the computation of the reciprocal series of a serie $S \in X\mathbb{K}[[X]]^*$, that is the series $\tilde{S} \in X\mathbb{K}[[X]]^*$ such that $S \circ \tilde{S} = X$. We did not pursue further this investigation as Example 7 below shows that the reverse role strategy fails in general - even assuming fast inversion formula. At minima, [17, Theorem 4.4] shows that computing the *characteristic monomials* of the Puiseux series of F centered at $(0,0)$ assuming that those of $\tilde{F}$ are given fits in the aimed bound. This data is of particular importance as it allows to compute the topological type of the branches of the germ of curve defined by F at $(0,0)$.

We could hope that there is always such a nice way to choose a suitable system of local coordinates in order to compute all the Puiseux series centered at $(0,0)$ - or at least their characteristic monomials - in less than cubic complexity in the total degree. Unfortunately, Example 7 below shows that this is hopeless. With the notations above, we have $\delta_H = \mu + n_Y - 1$ and $\delta_{\tilde{H}} = \mu + n_X - 1$ thanks to [37, Chapter II, Proposition 1.2, page 317], with $n_Y := \deg_Y(H) = \upsilon_Y(F(0,Y))$, $n_X := \deg_Y(\tilde{H}) = \upsilon_X(F(X,0))$ and $\mu := (F_X, F_Y)_0$ the Milnor number of the germ of curve defined by F at the origin. Thanks to the inversion formula, computing (the characteristic monomials of) at least half of the Puiseux series *centered at* $(0,0)$ with RNP3 while allowing the reverse role strategy costs $\Theta(\mu \min(d_Y/n_Y, d_X/n_X))$. Unfortunately, this can be $\Theta(D^3)$:

Example 7. Let $d > 6$ be divisible by 6 and let $F = (\phi + X^{d/2})^2 - \phi^{d/3}$ with $\phi = Y^3 - X^2$. So F has total degree $D = d$. We have $F_X = X\left((d\,X^{d/2-1} - 4)\,(\phi + X^{d/2}) + \frac{2d}{3}\,\phi^{d/3-1}\right)$ and $F_Y = Y^2\left(6\,(\phi + X^{d/2}) - d\,\phi^{d/3-1}\right)$. As $d \geq 12$, we have $(X, 6\,(\phi + X^{d/2}) - d\,\phi^{d/3-1})_0 = 3$, $(Y, U\,(\phi + X^{d/2}) + \frac{2d}{3}\,\phi^{d/3-1})_0 = 2$. We also have

$$\begin{array}{cl} & \left((3\,d\,X^{d/2-1} - 12)\,(\phi + X^{d/2}) + 2\,d\,\phi^{d/3-1}, 6\,(\phi + X^{d/2}) - d\,\phi^{d/3-1}\right)_0 \\ = & \left(3\,d\,X^{d/2-1}\,(\phi + X^{d/2}), 6\,(\phi + X^{d/2}) - d\,\phi^{d/3-1}\right)_0 \\ = & 3\,(d/2-1) + \left((\phi + X^{d/2}), \phi^{d/3-1}\right)_0 = -3 + d^2/2 \end{array}$$

We finally get $\mu = (F_X, F_Y)_0 = 6 + d^2/2 \in \Theta(d^2)$. Since $n_Y = 6$ and $n_X = 4$ we obtain $\min(d_Y \mu/n_Y, d_X \mu/n_X) = d^3/12 + d \in \Theta(d^3) = \Theta(D^3)$. The reverse role strategy is thus not helpful in that case.

More generally the Milnor number is invariant under local diffeomorphic change of coordinates $\pi : (\mathbb{K}^2, 0) \to (\mathbb{K}^2, 0)$. In Example 7, we can check that we always have $\max(n_X(\pi^* F), n_Y(\pi^* F)) = \max(n_X, n_Y)$, and - assuming π polynomial - we check further that we always have $\min(\deg_X(\pi^* F), \deg_Y(\pi^* F)) \geq \min(d_X, d_Y)$. Hence, there is no hope to reduce the polynomial F to a nicer polynomial G having faster desingularisation at $(0,0)$ (or even faster irreducibility test) using polynomial diffeomorphism of $(\mathbb{K}^2, 0)$ before appyling RNP3. This shows that our complexity results are sharp, and so independently of the choice of a polynomial local change of coordinates in $(\mathbb{K}^2, 0)$.

Note that this example is particularly sparse, but one could for instance consider the "dense" polynomial $F = Y^{d/3} + \sum_{k=0}^{d/6-1} (\phi + X^{d/2})^2\,\phi^k$ that will lead to the same conclusion than the one of Example 7.

References

[1] S. Abhyankar. *Algebraic Geometry for Scientists and Engineers*, volume 35 of *Mathematical surveys and monographs*. Amer. Math. Soc., 1990.

[2] S. S. Abhyankar. Irreducibility criterion for germs of analytic functions of two complex variables. *Advances in Mathematics*, 74(2):190 – 257, 1989.

[3] P. Alvandi, M. Ataei, and M. Moreno Maza. On the extended hensel construction and its application to the computation of limit points. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 13–20, New York, NY, USA, 2017. ACM.

[4] J.-D. Bauch, E. Nart, and H. Stainsby. Complexity of the OM factorizations of polynomials over local fields. *LMS Journal of Computation and Mathematics*, 16:139–171, 2013.

[5] D. Bini and V. Y. Pan. *Polynomial and Matrix Computations*, volume 1 of *Progress in Theoretical Computer Science*. Birkhäuser, Saarbrücken, 1994.

[6] A. Bostan, F. Chyzak, B. Salvy, G. Lecerf, and E. Schost. Differential equations for algebraic functions. In *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*, ISSAC '07, pages 25–32, 2007.

[7] R. P. Brent, F. G. Gustavson, and D. Y. Yun. Fast solution of toeplitz systems of equations and computation of padé approximants. *Journal of Algorithms*, 1(3):259 – 295, 1980.

[8] E. Brieskorn and H. Knörrer. *Plane Algebraic Curves*. Birkhaüser, 1986.

[9] D. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1990.

[10] J. Cassel. *Local Fields*, volume 3 of *Student Texts*. LMS, 1986.

[11] C. Chevalley. *Introduction to the Theory of Algebraic Functions of One Variable*, volume 6 of *Mathematical Surveys*. AMS, 1951.

[12] X. Dahan, E. Schost, M. M. Maza, W. Wu, and Y. Xie. On the complexity of the D5 principle. *SIGSAM Bull.*, 39(3):97–98, 2005.

[13] J. Della Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *EUROCAL 85*. Springer-Verlag LNCS 204, 1985.

[14] D. Duval. Rational Puiseux expansions. *Compositio Math.*, 70(2):119–154, 1989.

[15] D. Duval and A. Poteaux. Death of marc rybowicz, aged 52. *ACM Commun. Comput. Algebra*, 50(4):191–191, Feb. 2017.

[16] M. Eichler. *Introduction to the Theory of Algebraic Numbers and Functions*. Academic Press, 1966.

[17] E. R. García Barroso, P. D. González Pérez, and P. Popescu-Pampu. Variations on inversion theorems for newton–puiseux series. *Mathematische Annalen*, 368(3):1359–1397, Aug 2017.

[18] J. v. z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 3rd edition, 2013.

[19] X. Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *Journal of Complexity*, 14(2):257 – 299, 1998.

[20] F. Kako and T. Sasaki. Solving multivariate algebraic equations by Hensel construction. *Japan J. of Industrial and Applied Math.*, 16:257–285, 1999.

[21] E. Kaltofen. Greatest common divisors of polynomials given by straight-line programs. *J. ACM*, 35(1):231–264, Jan. 1988.

[22] H. T. Kung and J. F. Traub. All algebraic functions can be computed fast. *Journal of the Association for Computing Machinery*, 25(2):245–260, 1978.

[23] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[24] G. Moroz and É. Schost. A fast algorithm for computing the truncated resultant. In *ISSAC '16: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 1–8, New York, NY, USA, 2016. ACM.

[25] D. R. Musser. Multivariate polynomial factorization. *J. ACM*, 22(2):291–308, Apr. 1975.

[26] B. M. T. O. Cormier, M. F. Singer and F. Ulmer. Linear differential operators for polynomial equations. *J. Symbolic Comput.*, 34(5):355–398, 2002.

[27] C. Pascal and E. Schost. Change of order for bivariate triangular sets. In *ISSAC'06*, pages 277–284. ACM, 2006.

[28] J. M. Peral. *Polígonos de newton de orden superior y aplicaciones aritméticas*. PhD thesis, Universitat de Barcelona, 1999.

[29] A. Poteaux. *Calcul de développements de Puiseux et application au calcul de groupe de monodromie d'une courbe algébrique plane*. PhD thesis, Université de Limoges, 2008.

[30] A. Poteaux and M. Rybowicz. Good reduction of puiseux series and complexity of the newton-puiseux algorithm over finite fields. In *ISSAC '08: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 239–246, New York, NY, USA, 2008. ACM.

[31] A. Poteaux and M. Rybowicz. Complexity bounds for the rational newton-puiseux algorithm over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 22:187–217, 2011. 10.1007/s00200-011-0144-6.

[32] A. Poteaux and M. Rybowicz. Good reduction of puiseux series and applications. *Journal of Symbolic Computation*, 47(1):32 – 63, 2012.

[33] A. Poteaux and M. Rybowicz. Improving complexity bounds for the computation of puiseux series over finite fields. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, pages 299–306, New York, NY, USA, 2015. ACM.

[34] A. Poteaux and E. Schost. On the complexity of computing with zero-dimensional triangular sets. *Journal of Symbolic Computation*, 50(0):110 – 138, 2013.

[35] A. Schönage and V. Strassen. Schnelle multiplikation großer zahlen. *Computing 7*, pages 281–292, 1971.

[36] V. Shoup. Fast construction of irreducible polynomials over finite fields. *Journal of Symbolic Computation*, 17:371–391, 1994.

[37] B. Teissier. Cycles évanescents, sections planes et conditions de whitney. In *Singularités à Cargèse*, number 7-8 in Astérisque, pages 285–362. Société mathématique de France, 1973.

[38] J. Van Der Hoeven and G. Lecerf. Accelerated tower arithmetic. Preprint, 2018.

[39] R. J. Walker. *Algebraic Curves*. Springer-Verlag, 1950.

[40] M. Weimann. Bivariate factorization using a critical fiber. *Journal of Foundations of Computational Mathematics*, pages 1–45, 2016.